

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



TRABAJO FIN DE GRADO

**Diseño simplificado de videojuegos
para educación infantil basada en una
herramienta de autor**

Juan Covas Medina

Enero 2015

Diseño simplificado de videojuegos para educación infantil basada en una herramienta de autor

AUTOR: Juan Covas Medina
TUTOR: David Camacho Fernandez

Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid

Enero de 2015

Agradecimientos

Quiero dar gracias al Centro Mater Inmaculata, que es el centro donde estudié, por permitirme hacer las pruebas y darme toda la ayuda necesaria para poder elaborar este trabajo. A todos los profesores que han colaborado con sus opiniones. En especial a la pedagoga del colegio, una amiga junto a la que he pasado muchas cosas y tuve la oportunidad de reencontrarme en un grata sorpresa al encontrarla trabajando en el colegio. A la coordinadora y profesora de infantil que ha colaborado directamente en las pruebas, que habiendo sido mi profesora hace ya 23 años, aun recuerda mi nombre y mi apellido. Simplemente increíble. También quiero agradecer y mencionar en particular a los alumnos que han realizados las pruebas por su colaboración, chicos diestros en el manejo de la pizarra digital y muy cariñosos.

Gracias a mi tutor David Camacho por permitirme hacer un TFG que me guste, que pueda disfrutar, me motive. Considero que es importante terminar el grado haciendo algo que te guste y él me lo ha permitido. Gracias también por aguantarme durante la elaboración y el desarrollo y darme parte del poco tiempo que tienes.

A amigos y familia en general. En particular a un amigo, compañero de carrera y muchas prácticas. Sin sus empujones en la dirección correcta creo que nunca hubiera llegado hasta aquí.

Por ultimo gracias a mi padre que me ha apoyado durante toda la carrera y ha sabido hacerme valorar la importancia de estudiar y formarse, no solo por tener un título, sino por lo que te aporta como persona conocer y saber hacer cosas nuevas que antes no conocías y como es posible enfrentarte a nuevos retos, resolver más problemas y poder dirigir mejor tu vida gracias a ese conocimiento

Resumen

Las pizarras digitales son uno de los grandes avances de los últimos tiempos y clave en la innovación docente actual. Bien utilizada, es capaz de ayudar a los profesores a enseñar a los niños de una manera más amena, y que llame la atención de estos para que aprendan divirtiéndose. En esto se fundamentan nuevas técnicas de enseñanza como la gamificación, que es el uso de técnicas y dinámicas propias de los juegos y el ocio en actividades no recreativas.

Para que este avance sea útil es necesario que exista una herramienta y un repositorio de recursos para poder crear contenido que permita aprovechar las pizarra digitales en el aula. Hablando con profesores de diferentes edades y capacidades informáticas se llegó al a conclusión de es que existen algunos recursos pero son confusos, limitados, o difíciles de entender y manejar tanto por los profesores como por los alumnos. Por otra parte, la aparición de *tablets*, que tan intuitivas resultan para los niños y que utilizan una interacción equivalente a las pizarras digitales , abre la posibilidad de extender este tipo de aprendizaje basado en juegos desde la pizarra a casa. Con este proyecto se ha tratado de solucionar esos problemas, creando una herramienta de autor, pensada y orientada para que los profesores puedan crear, editar y almacenar recursos educativos fácilmente, para mejorar e incentivar el uso de las pizarras digitales en las aulas, así como intentar llevar este tipo de juegos educativos a las *tablets*. La herramienta ha sido implementada en *HTML5+JavaScript*, de esta manera se logra la compatibilidad de la herramienta con diferentes tipos de dispositivos y sistemas operativos haciendo uso del navegador como plataforma de ejecución. Aprovechando las ventajas de *HTML5*, se ha creado una herramienta que funciona totalmente en local, sin necesidad de conexión a Internet, y que, haciendo uso de la exportación de archivos en formato json permite compartir juegos.

Finalmente será posible ejecutar la aplicación desde un servidor para jugar a los juegos desde las *tablets* donde no es posible ejecutar dicha aplicación haciendo uso del sistema local de archivos.

Para evaluar la funcionalidad y usabilidad de la herramienta hemos contado con la colaboración del colegio Mater Immaculata que nos ha dejado realizar varias pruebas en las aulas de Infantil, y con la opinión de varios profesores para la correcta evaluación de la aplicación.

Abstract

Digital whiteboards are one of the greatest advances in recent times, key for actual teaching innovation. When properly used, they help teachers to teach children in a more entertaining way, attracting their attention to learn while having fun. On these grounds new teaching techniques arise, such as gamification, which is applying techniques and dynamics from games and leisure to non-recreational activities.

For this advance to be useful, a Tool and a Resources Repository are necessary to enable easily create content that will make possible to take advantage of digital whiteboards within classrooms. Conversations with professors of different ages and computer knowledge, allowed to realize that some resources do exist, but disorganized, limited or difficult to understand and maintain for pupils and teachers alike.

On the other hand, the appearance of *tablets*, which are so intuitive to children and equally interactive as digital whiteboards are, opens the possibility to extend this type of learning based on games from whiteboards to home.

These are the problems this project intends to resolve, by creating a tool for authors, thought and oriented to enable the teachers to create, edit and store educational resources easily, to improve and encourage the use of digital whiteboards in class, as well as to take this kind of educational games to *tablets*.

The tool is implemented in HTML5+*JavaScript*, which allows it to be compatible with different devices, operating systems, and browsers used as the platform for its execution. Leveraging on HTML5 advantages a fully functional locally tool has been created, without Internet connection, and able to share games by exporting Json format files.

Finally, it is possible to run the application from a server to play games from *tablets* where the application cannot be executed by making use of local file system.

To evaluate tool's functionality and usage, we have counted on the collaboration of the Mater Immaculata school, which has let us to do several tests in its Infant classrooms and gather several teacher's opinions to correctly evaluate the application.

INDICE DE CONTENIDOS

1 Introducción	1
1.1 Motivación	1
1.2 Objetivos	2
1.3 Organización de la memoria	2
2 Estado del arte.....	3
2.1 Introducción	3
2.2 Aplicaciones estudiadas antes del desarrollo.	3
2.3 Conclusiones	6
3 Diseño	7
3.1 Introducción	7
3.2 Análisis previo	7
3.3 Tecnologías usadas.....	8
3.3.1 HTML5	8
3.3.1.1 Canvas	9
3.3.1.2 IndexedDB	9
3.3.2 JavaScript	10
3.3.3 JSON	11
3.4 Diseño de la Herramienta.....	13
3.4.1 El patrón MVC.....	13
3.4.2 La Vista	15
3.4.3 El Modelo.....	18
3.4.4 El Controlador.....	20
3.4.5 Las plantillas	21
4 Desarrollo.....	22
4.1 Introducción	22
4.2 Estructura de archivos del proyecto	22
4.3 Vista..	23
4.4 El modelo	29
4.5 El controlador.....	32
4.6 Las Plantillas	35
4.7 La aplicación para tablets.....	36
5 Integración, pruebas y resultados	37
5.1 Introducción	37

5.2 Pruebas de Integración	37
5.3 Resultados de la prueba de integración.....	38
5.4 Pruebas de usabilidad.....	38
5.4.1 Prueba para los profesores	38
5.4.2 Resultado de la prueba para profesores.....	39
5.4.3 Pruebas para alumnos.....	42
5.4.4 Resultado de las pruebas para alumnos.....	42
6 Conclusiones y trabajo futuro.....	45
6.1 Conclusiones	45
6.2 Trabajo futuro.....	45
Referencias.....	47
Glosario.....	49
Anexos	I
A Manual de Usuario	I
B Informe de la coordinadora de Infantil.	XV

INDICE DE FIGURAS

Figura 2-1 Ejemplo de juego en flash.....	4
Figura 2-2 Ejemplo de juego configurable.....	5
Figura 2-3 Ejemplo de juego de ordenación de letras.....	5
Figura 3-1 Esquema del patrón MVC.....	14
Figura 3-2 Navegación por las pantallas de la herramienta.....	15
Figura 3-3 Esquema de la página de inicio.....	16
Figura 3-4 Esquema de una plantilla para generar juegos.....	16
Figura 3-5 Navegación por las pantallas de la herramienta.....	17
Figura 3-6 Esquema de los elementos componen un canvas	18
Figura 3-7 Esquema de la interacción entre los componentes de la aplicación.	20
Figura 4-1 Esquema del directorio de la aplicación.....	23
Figura 4-2 Pagina de inicio de la aplicación	25
Figura 4-3 Plantilla de la aplicación con sus diferentes acciones	26
Figura 4-4 Sistema de ayudas	27
Figura 4-5 Página de juego.....	29
Figura 5-1. Primer Juego de referencia.....	39
Figura 5-2. Resultado del primer juego.....	39
Figura 5-3. Juego de ejemplo de relaciones.....	40
Figura 5-4. Resultado del segundo juego creado.....	40
Figura 5-5. Juego de cuentas matemáticas creado.....	41

Figura 5-6. Resultado de la encuesta a niños	42
Figura A-1 Pantalla de inicio de la aplicación.....	I
Figura A-2 Esquema de una plantilla.....	II
Figura A-3 Ejemplo de sistema de ayudas.....	III
Figura A-4 Plantilla 1 de la aplicación.....	III
Figura A-5 Ejemplo de juego de plantilla 1.....	IV
Figura A-6 Plantilla 2 de la aplicación.....	V
Figura A-7 Ejemplo de juego de plantilla 2.....	VI
Figura A-8 Plantilla 3 de la aplicación.....	VII
Figura A-9 Ejemplo de juego de plantilla 3.....	VIII
Figura A-10 Plantilla 4 de la aplicación.....	IX
Figura A-11 Ejemplo de juego de plantilla 4.....	X
Figura A-12 Plantilla 5 de la aplicación.....	XI
Figura A-13 Ejemplo de juego de plantilla 5.....	XII
Figura A-14 Opciones.....	XIII
Figura A-14 Pantalla de inicio en la versión de servidor.....	XIII
Figura A-14 Opciones de carga de archivos en el servidor.....	XIV

INDICE DE TABLAS

Tabla 5-1 Resultados de compatibilidad en navegadores.....	39
Tabla 5-2 Preguntas de la encuesta a la profesora.....	43
Tabla 5-3 Preguntas sobre la reacción de los niños.....	45

1 Introducción

1.1 Motivación

Dado el avance de la tecnología en la actualidad se ha vuelto imprescindible que está sea introducida en las aulas y en el sistema educativo como medio para el aprendizaje. Hace ya algunos años dicha introducción pasaba por aulas de informática orientadas a aprender a usar los ordenadores. En la actualidad el siguiente paso ha sido usar las nuevas herramientas tecnológicas para la enseñanza de las materias generales. En particular se está extendiendo el uso de pizarras digitales en las aulas y por tanto es necesario disponer del software necesario para enseñar usándolas. Conociendo dicha situación se ha decidido desarrollar una herramienta para que los profesores puedan crear y adaptar los juegos y pruebas, presentes habitualmente en cuadernos o medios similares, a estas nuevas pizarras. Esta herramienta está orientada de forma particular al temario y necesidades de la educación de nivel infantil.

Además haciendo uso de HTML5 y JavaScript facilitamos que la herramienta se pueda utilizar fácilmente desde los navegadores de *tablets* y ordenadores para trasladar este tipo de juegos a los hogares de los niños y homogeneizando en cierta medida las actividades para el aprendizaje en casa y en las aulas.

A parte de esta compatibilidad se ha hecho lo posible para que la aplicación sea de fácil uso para padres y profesores que en muchos casos tienen dificultades para manejar herramientas informáticas, aunque esa situación se va de cada vez menos.

1.2 Objetivos

El objetivo de trabajo es realizar una herramienta de autor que permita a los profesores partiendo de un conjunto de plantillas adaptar los ejercicios que habitualmente se realizan en el nivel de educación infantil a un versión generada por la herramienta y que se pueda usar en las pizarras digitales de las aulas y en los ordenadores y *tablets* que muchos alumnos tienen en sus casas.

La herramientas actuales para pizarras digitales son muy limitadas y complejas de usar para muchos profesores, por ello es importante desarrollar una interfaz sencilla y clara con suficientes ejemplos que puedan orientar a profesores y padres en el desarrollo de nuevos juegos partiendo de las plantillas y ejemplos.

1.3 Organización de la memoria

Este documento describe las diferentes fases que se han realizado para el desarrollo del proyecto, empezando por el apartado 2 en el que realizaremos un estudio del estado del arte, en el cual analizamos algunas herramientas, similares a la que se quiere implementar, analizando sus cosas positivas y las posibles cosas negativas, que podemos aprovechar para la realización de nuestro proyecto.

En el apartado 3 se explica el diseño de la aplicación, cómo están organizados sus diferentes módulos, las relaciones entre ellos, los patrones de diseño usados y las funcionalidades importantes utilizadas de las diferentes tecnologías usadas para el desarrollo.

En el apartado 4 comentaremos en detalle los puntos clave de la implementación de la aplicación, haciendo hincapié en los diferentes obstáculos que se plantearon durante el desarrollo y las soluciones que propuestas para sortearlos.

En el apartado 5 detallaremos las pruebas realizadas tanto en las aulas, como con los profesores y alumnos, como las pruebas de compatibilidad e integración.

Por último en el apartado 6 comentaremos las conclusiones finales, y estudiaremos si se han cumplido los objetivos propuestos y que motivan el desarrollo de la herramienta.

2 Estado del arte

2.1 Introducción

Antes de comenzar a realizar el proyecto, se hizo una labor de investigación para analizar las herramientas que ya existen con el mismo o similar objetivo que el de este trabajo. Esta tarea nos ayuda a tener una perspectiva de que es factible conseguir y que fallos y utilidades tienen los recursos existentes actuales para usarlos de referencia.

2.2 Aplicaciones estudiadas antes del desarrollo.

No ha sido fácil encontrar ejemplos que se aproximen a las ideas que buscamos sin embargo se han encontrado algunos ejemplos que pueden ser útiles de analizar.

Buscando en internet Juegos para pizarra digital y términos similares se encontró la página <http://recursospdj.wordpress.com/> que enlaza varios juegos y recursos en principio compatibles con pizarras.

Navegando en ella se encontraron varios ejemplos de páginas con actividades que podrían ejecutarse sobre pizarras digitales. No se encontraron enlaces orientados a niños del nivel de infantil y muchos de ellos eran enlaces que ya no funcionaban, sin embargo había algún enlace que lo hacía, como por ejemplo, un juego de cálculo mental contrarreloj. El problema encontrado, bastante sorprendente, fue que la entrada del resultado era por teclado lo que lo convertía en un recurso inapropiado para las pizarras digitales. Sin embargo tras posteriores reflexiones se razonó que cualquier recurso escrito necesitaría de un teclado táctil. (aunque en principio se proporciona con el software de las pizarras).



Figura 2-1 Ejemplo de juego en flash

El segundo lugar donde se encontraron ejemplos , esta vez más cercanos a lo esperado para juegos de infantil , fue en www.edicioneslolapirindola.com

En esta página había un conjunto de recursos para educación infantil y entre ellos recursos interactivos que por su características de interacción eran viables para ser usados en pizarras digitales. Había varios juegos clásicos configurables como Cartas de Memoria

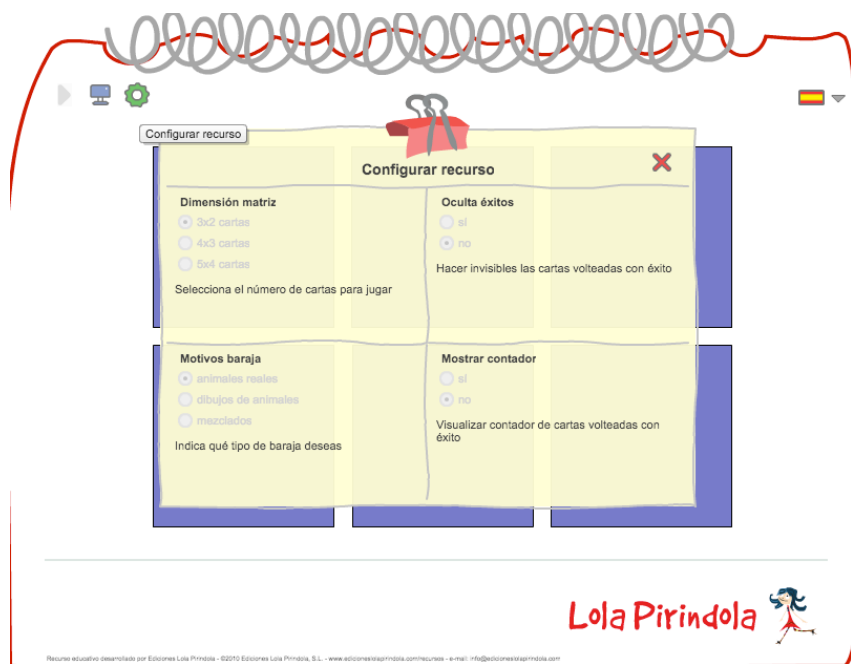


Figura 2-2 Ejemplo de juego configurable

Otro juego en apariencia apropiado para niños de infantil trataba sobre la ordenación de letras desordenadas para formar una palabra como se ve en la siguiente imagen.

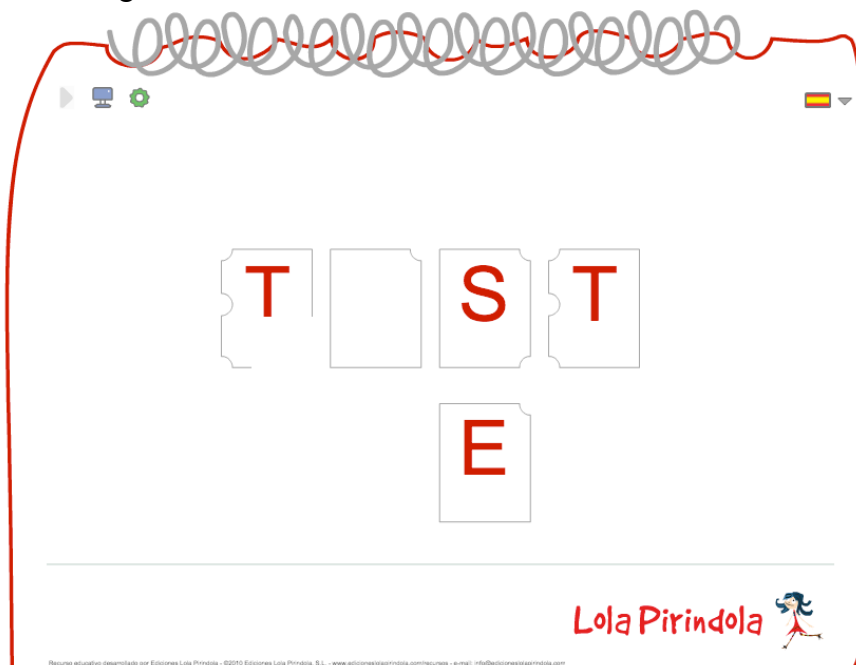


Figura 2-3 Ejemplo de juego de ordenación de letras

Ambos ejemplos podían configurarse para crear múltiples juegos pero requerían de suscripción a la plataforma para hacerlo posible.

Búsquedas posteriores acabaron en la mayor parte de los casos en paginas de editoriales para libros de aprendizaje para niños con algunos juegos en flash pero en ningún caso con la posibilidad de modificarlos.

Actualmente son esta clase de recursos cerrados los que se están utilizando en las aulas con pizarras digitales.

2.3 Conclusiones

No hay herramientas en este momento apropiadas para el desarrollo de juegos con los objetivos propuestos.

En el apartado técnico lo normal es encontrar juegos desarrollados en Flash lo cual suele llevar a varios problemas de compatibilidad entre navegadores y aun que en muchos casos compatibles con pizarras digitales no han sido desarrollados pensando en estas.

En lo docente hemos podido comprobar con ayuda de los profesores que el contenido de los ejemplos y recursos no se ajustan por lo general a las necesidades que presentan el nivel de educación infantil

Dado que no hay ningún ejemplo claro de herramienta que haya sido desarrollada para los profesores y que garantice que el contenido se ajusta a las necesidades de lo que se aprende en las aulas, y sea compatible la tecnología que se usa, se puede concluir que la herramienta puede efectivamente ser útil para cubrir una necesidad del sistema educativo en lo referente a las nuevas tecnologías.

En los siguientes apartados de hablará del desarrollo de dicha herramienta que se espera mejoré técnicamente lo existente y garantice la calidad de los contenidos estableciendo plantillas ,ejemplos y forma de uso, en función de a la información y consejos del personal docente.

3 Diseño

3.1 Introducción

En este apartado detallaremos el proceso de diseño de la aplicación. Para desarrollar una herramienta de autor apropiada es necesario tener en cuenta las cuestiones mencionadas en la motivación del proyecto , hacer un análisis previo y tomar las decisiones apropiadas en cuanto a la tecnología a usar y que requisitos satisfacer para conseguir desarrollar una herramienta potente y útil .

3.2 Análisis previo

Estudiando el contexto en el que se usaría principalmente la aplicación tenemos como lugar principal el colegio y las aulas y potencialmente en la casas para lograr cumplir con el objetivo de que los niños puedan jugar en el aula y en casa a los mismos juegos.

Tras analizar la tecnología disponible en estos lugares y preguntar a profesores sobre sus necesidades hemos obtenido un conjunto de requisitos que sirven como guía para decidir un diseño a usar en el desarrollo de la herramienta:

- 1- El herramienta debe poder usarse en la mayor variedad de sistemas operativos posibles.
- 2- Los juegos deben ejecutarse sin problema en pizarras digitales donde se hará su uso principal y a ser posible en ordenadores y *tablets* para extender su uso a los hogares.
- 3- La herramienta necesita ser de fácil instalación y debería poder utilizarse de manera local, lo que implica tener un sistema de persistencia.
- 4- Los juegos desarrollados debería poder exportarse e importarse para poder ser compartidos entre profesores y padres.
- 5- La interfaz tiene que ser sencilla e intuitiva para que sea de fácil uso
- 6- Los juegos deberían ser apropiados para la edad de los niños por lo que las plantillas deben adaptarse y basarse en los recursos que se usan actualmente en los libros de enseñanza para infantil.
- 7- La herramienta necesitará un conjunto de plantillas y juegos de ejemplo y iniciales que facilite el aprendizaje de los profesores y el desarrollo de nuevos juegos.

3.3 Tecnologías usadas

Una vez detallados los requisitos, se explican a continuación las características y ventajas de las tecnologías que se han escogido para desarrollar la aplicación y como permitirán cumplir con dichos requisitos.

3.3.1 HTML5

HTML5 es un lenguaje moderno de etiquetas, el cual todavía está en estado de desarrollo. Este lenguaje nació cuando se unieron los dos grupos de desarrolladores que estaban intentando actualizar el lenguaje HTML4, WHATWG (iniciales de Web Hypertext Application Technology Working Group) que estaba desarrollando una especificación mejor del lenguaje HTML y el W3C (World Wide Web Consortium) que en ese momento estaba desarrollando la versión 2.0 del lenguaje XHTML. Gracias a la unión de estos dos equipos se abandonaron los desarrollos independientes y se centraron en el desarrollo del lenguaje HTML5, la inclusión del W3C asegura que el lenguaje HTML5 será un estándar gratuito.

HTML 5 implementa nuevas funcionalidades para reducir la dependencia de *plugins* externos cuyo ejemplo más relevante es el de Adobe Flash, con el cual se han introducido el video, sonido y juegos en las páginas web hasta ahora. Ante el obvio aumento de webs que muestran información en audio y video y que experimentan con juegos sobre navegador, HTML5 establece 3 etiquetas nuevas que permiten realizar webs con dicha funcionalidad sin depender de *plugins* como Flash.

<video>: Permite añadir videos en la página sin necesidad de tener un reproductor externo para poder visualizarlos.

<audio>: Esta etiqueta permite añadir audio capaz de reproducirse sin necesidad de “*plugins*” externos al navegador.

<canvas>: Añade un elemento en el cual se pueden incluir dibujos y animaciones, de esta manera se evita la dependencia de Flash.

Se ha hecho uso del audio y canvas para la ejecución de juegos en la aplicación. Hablaremos de este último elemento en profundidad más adelante.

HTML5 también establece un sistema de persistencia sobre el navegador denominado *IndexedDB* que permite guardar datos de manera local en el navegador usando una base de datos indexada. La aplicación usará este sistema de persistencia como se detallará más adelante.

3.3.1.1 Canvas

El canvas de HTML5 permite tener un espacio en la página donde permita dibujar elementos como algunas formas geométricas y alterar su color , aplicar gradientes y otros efectos relacionados con la tecnología de la creación de gráficos. Por supuesto también permite mostrar imágenes y texto lo cual resultará esencial para el desarrollo de la aplicación. Haciendo uso de el canvas y *JavaScript* se puede modificar los elementos dentro del primero para realizar operaciones gráficas y hacerlo reaccionar a eventos como los del ratón. Estas propiedades han permitido que se hayan podido implementar motores gráficos que funcionan sobre el canvas permitiendo así el desarrollo posterior de videojuegos.

Se hará uso de estas propiedades para crear un pequeño motor de juegos que permita manejar textos, imágenes y algunas figuras geométricas.

3.3.1.2 IndexedDB

IndexedDB[1] es un almacén de objetos. No es lo mismo que una base de datos relacional, que tiene tablas, con columnas y filas de colecciones. Es una diferencia fundamental e importante y afecta a la forma en la que se diseña y se crean las aplicaciones.

IndexedDB se diferencia en que crea un almacén de objetos de un tipo de datos y los objetos *JavaScript* simplemente se guardan en ese almacén. Cada almacén de objetos puede tener una colección de índices, lo que permite realizar consultas e iteraciones de forma eficiente.

IndexedDB también acaba con la noción de lenguaje de consultas estándar (Standard Query Language, SQL), que es sustituido por una consulta en un índice, lo que produce un cursor que puedes utilizar para iterar en el conjunto de resultados.

Esta tecnología es la recomendada oficialmente por el W3C para el almacenamiento de datos en el navegador.

3.4 JavaScript

JavaScript nació en 1995 por la necesidad de ejecutar acciones de manera local en el ordenador de los usuarios de páginas web [2]. Este requisito surge cuando las páginas, y aplicaciones web comenzaron a evolucionar hasta el punto de convertirse en aplicaciones Web que requerían el uso de formularios para enviar información al servidor. Esto derivó en problema de que se alargaran los tiempos de comunicación entre cliente y servidor debido los errores del usuario al rellenar datos y tener que esperar la respuesta de error del servidor. Por ese motivo nace *JavaScript* que permitía comprobar los datos de dichos formularios desde el lado del cliente minimizando el uso de la comunicación (en un momento en que dicha comunicación era relativamente lenta) permitiendo comprobar el formato de los datos desde el lado del cliente.

JavaScript es un lenguaje pensado para ejecutarse de manera local en el ordenador del usuario final, por lo que se han implementado ciertas restricciones para proteger al usuario de posibles ataques de aplicaciones web maliciosas y limitar los daños que pueda provocar el propio usuario debido a un uso indebido. Esto se consigue limitando el acceso de la aplicación al dominio en el que se está ejecutando, e impidiendo su acceso al disco duro del equipo. Tampoco permite la modificación de las preferencias del navegador ni de ventanas que no hayan sido abiertas por la aplicación.

Actualmente su uso se esta extendiendo Además, gracias a la aparición de HTML5 y a sus nuevos elementos, JavaScript se ha convertido en el complemento perfecto de este, para incrementar e implementar las funcionalidades que hemos comentado anteriormente. Por todo ello es muy apropiado para el desarrollo de la aplicación.

3.4.1 JSON

JSON (JavaScript Object Notation - Notación de Objetos de *JavaScript*)[3] es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación *JavaScript*, Standard ECMA-262 3rd Edition de Diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje y será el formato usara para guardar la información relativa a los juegos de la aplicación ya que es la forma natural de guardar objetos en *JavaScript* y aporta mucha flexibilidad para manejar dicha información en la aplicación

JSON está constituido por dos estructuras:

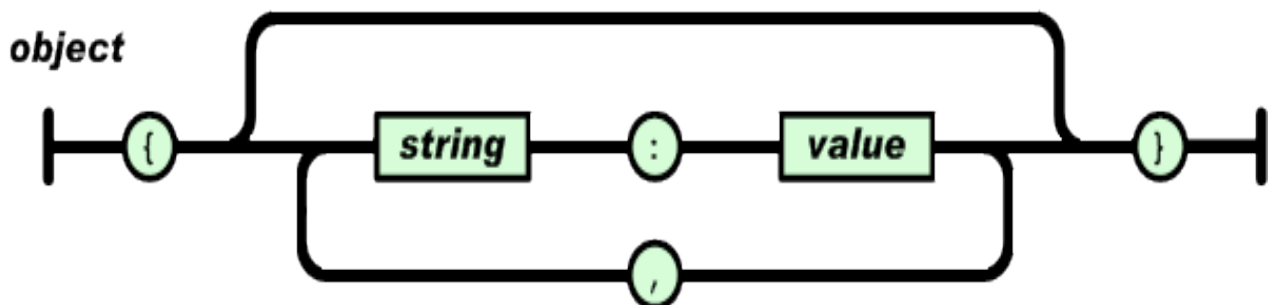
- Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.

- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

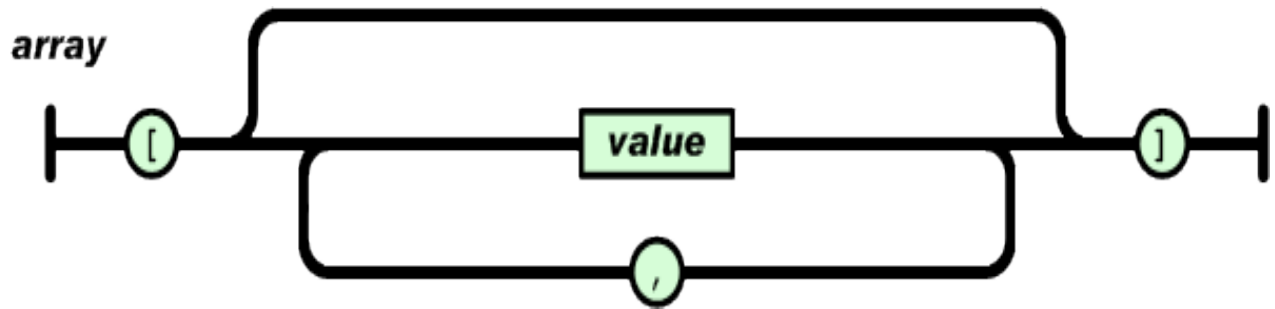
Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras.

En JSON, se presentan de estas formas:

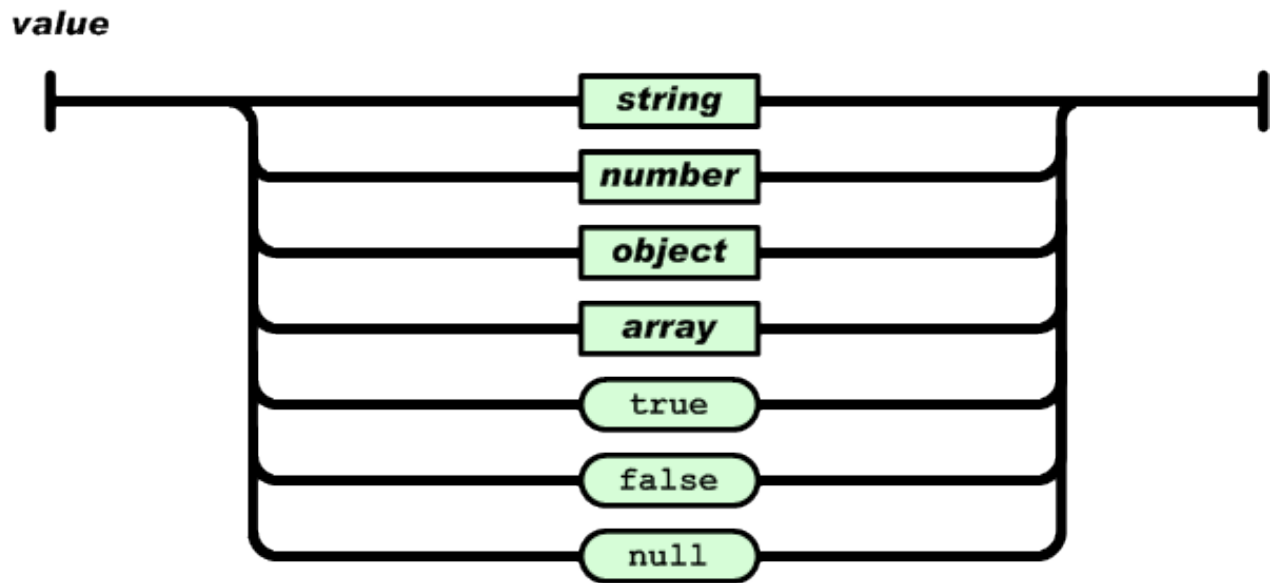
Un objeto es un conjunto desordenado de pares nombre/valor. Un objeto comienza con { (llave de apertura) y termine con } (llave de cierre). Cada nombre es seguido por : (dos puntos) y los pares nombre/valor están separados por , (coma).



Un arreglo es una colección de valores. Un arreglo comienza con [(corchete izquierdo) y termina con] (corchete derecho). Los valores se separan por , (coma).



Un valor puede ser una cadena de caracteres con comillas dobles, o un número, o true o false o null, o un objeto o un arreglo. Estas estructuras pueden anidarse.



3.5 Diseño de la Herramienta

Haciendo uso de las tecnologías anteriores ha sido posible desarrollar una aplicación web que se ejecuta en el lado del cliente y que al usar el navegador como base permite cumplir con el primer requisito de alta portabilidad ya que al usar estándares bien establecidos y altamente respetados ya por los navegadores su ejecución en distintos sistemas operativos a través del navegador nos garantiza la compatibilidad buscada en ordenadores y para la pizarra digital, y nos acerca a una solución compatible con *tablets*. Este tipo de aplicación permite cumplir con otro requisito, que la instalación sea fácil, ya que solo será necesario traspasar los ficheros de un ordenador a otro y ejecutar el .HTML que inicie en el navegador la aplicación

En los siguientes apartados se explica el patrón usado para el diseño de la herramienta y su uso en particular en este caso.

3.5.1 El patrón MVC

El modelo–vista–controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

De manera genérica, los componentes de MVC se podrían definir como sigue:

- **El Modelo:** Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.

- **El Controlador:** Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta de 'modelo' (por ejemplo, desplazamiento o scroll por un

documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.

- **La Vista:** Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.

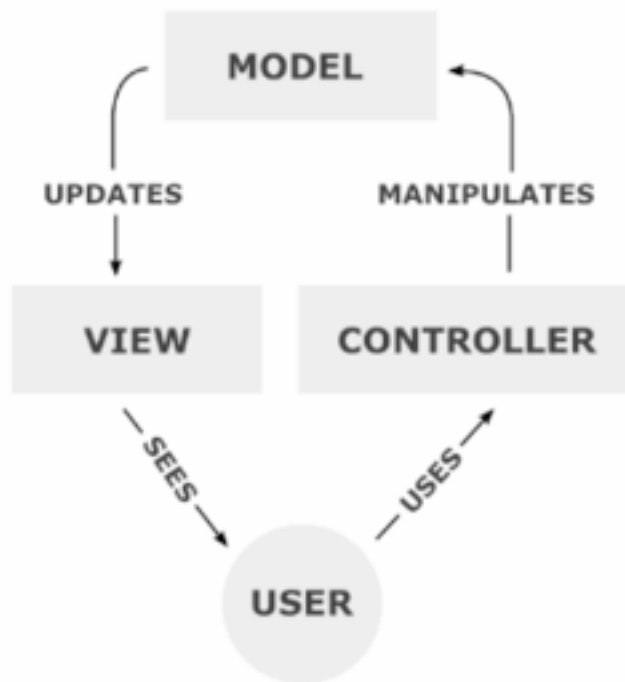


Figura 3-1 Esquema del patrón MVC

Utilizando este patrón se han diseñado los tres componentes de la aplicación se han diseñado la Vista el Modelo y el Controlador que se detallan a continuación.

3.5.2 La Vista

La vista se ha desarrollado usando la tecnología HTML para crear páginas que sirven como medio para desarrollar, modificar y usar los juegos y el uso de HTML y CSS va garantiza una homogeneidad de apariencia para los distintos sistemas operativos y navegadores.

Las distintas vistas de la aplicación se distribuirán de la siguiente manera:

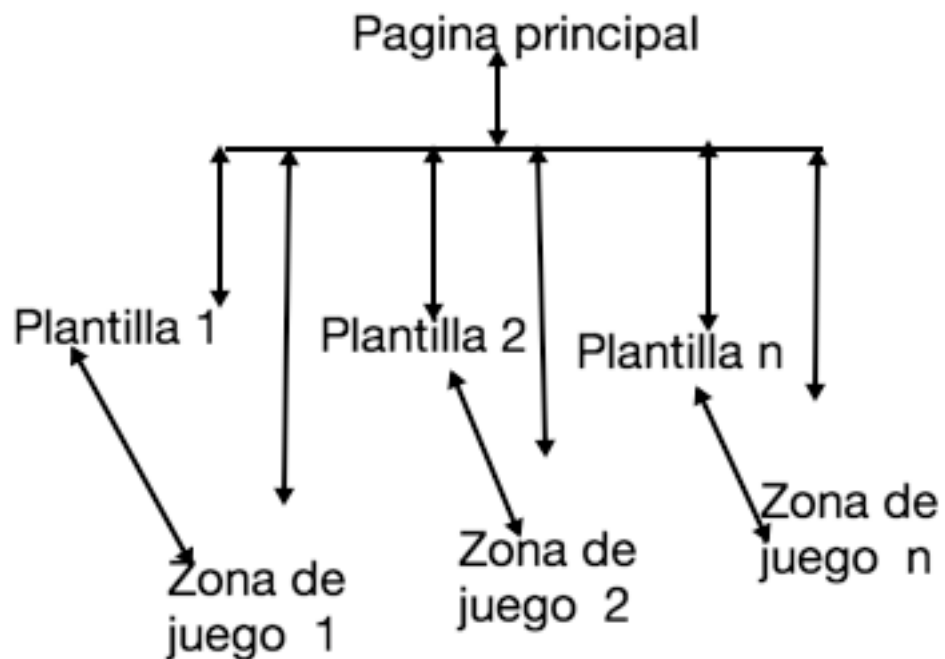


Figura 3-2 Navegación por las pantallas de la herramienta

Como se aprecia en la figura habrá una distribución sencilla de páginas. Una página inicial donde se muestran el tipo de juegos y los correspondientes enlaces que permitan ir a la página de la plantilla para modificar y crear juegos de ese tipo y otro enlace que nos lleve a la página para ejecutar el tipo de juego.

En la página principal tendremos una distribución de elementos aproximada como la que sigue

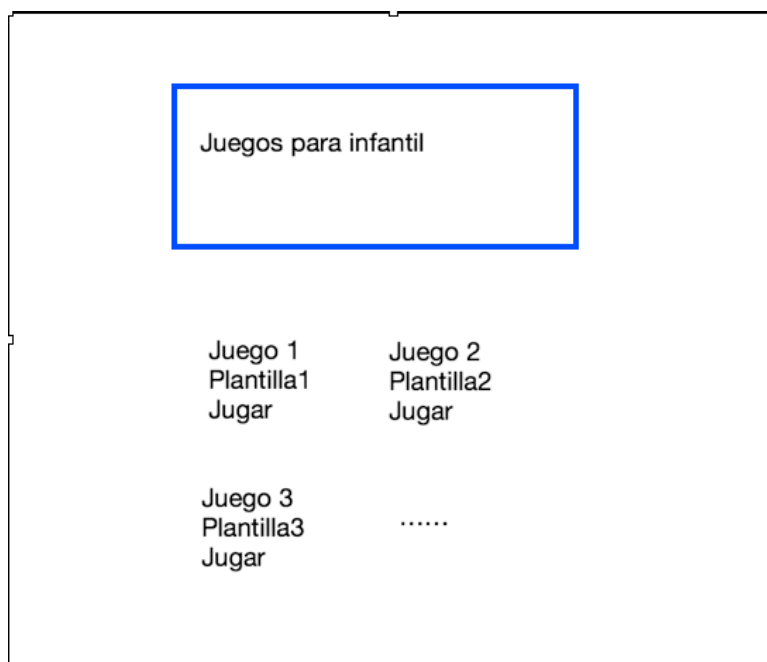


Figura 3-3 Esquema de la página de inicio

Se podría resumir esta pantalla como un directorio con los enlaces a las páginas donde ejecutar juegos y a las plantillas para crearlos y modificarlos.

Las plantillas tendrán una distribución como la siguiente como base:

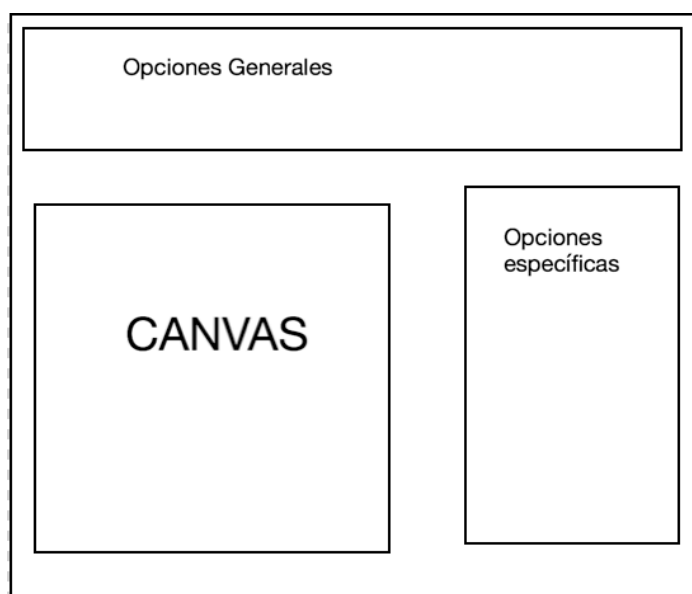


Figura 3-4 Esquema de una plantilla para generar juegos

En la cabecera tendrá una serie de opciones comunes a todas las plantillas:

- Establecer el título del juego
- Nombre con el que se guardará el juego
- Cambiar el tamaño del canvas.

Encontraremos a la izquierda, el canvas, donde se añadirán y modificarán los elementos del juego. Debajo hay un pie que marca la posición del ratón y a la derecha una serie de opciones que serán específicas de la plantilla, si bien un conjunto de ellas serán comunes, como por ejemplo, cambiar el tamaño de los elementos, insertar texto, imágenes, etc.

Las opciones específicas servirán para delimitar de alguna manera las acciones del usuario garantizando unas características que se correspondan con las necesidades y restricciones que tendrá la ejecución de ese tipo de juego.

La página de ejecución mostrará un canvas y la lista de juegos disponibles para cargar.

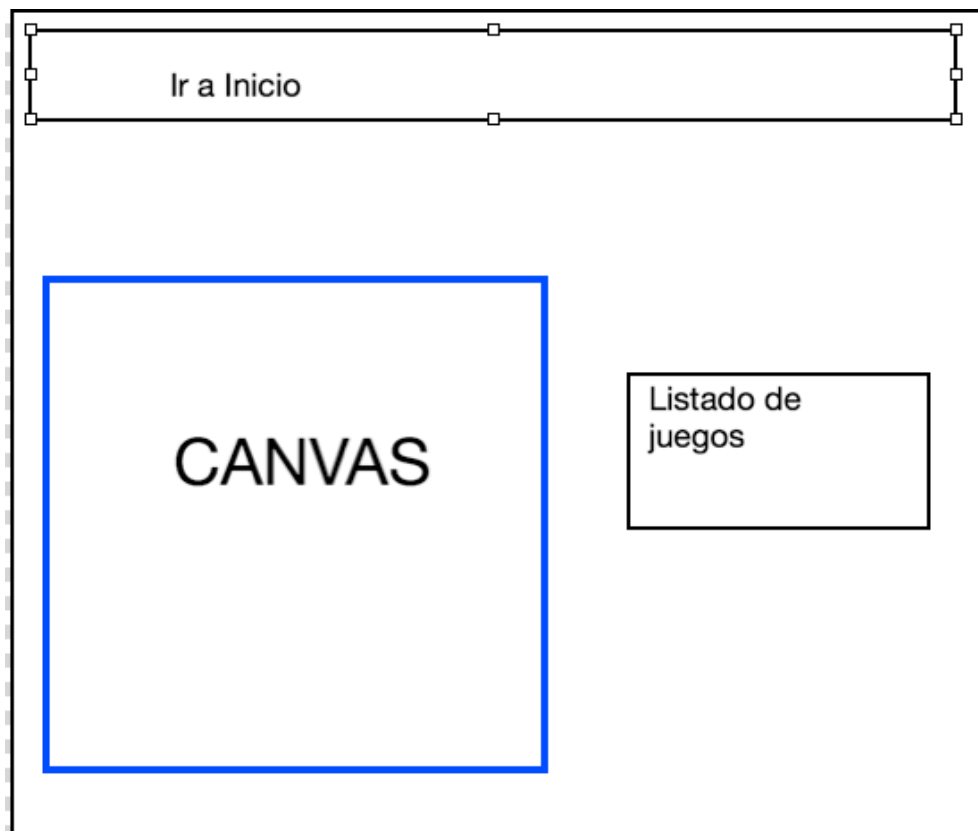


Figura 3-5 Esquema de una página para ejecutar los juegos.

3.5.3 El Modelo

Todos los juego están modelados como un canvas con un conjunto de elementos con los que interactuar.

Los elementos se modelan de la siguiente manera.

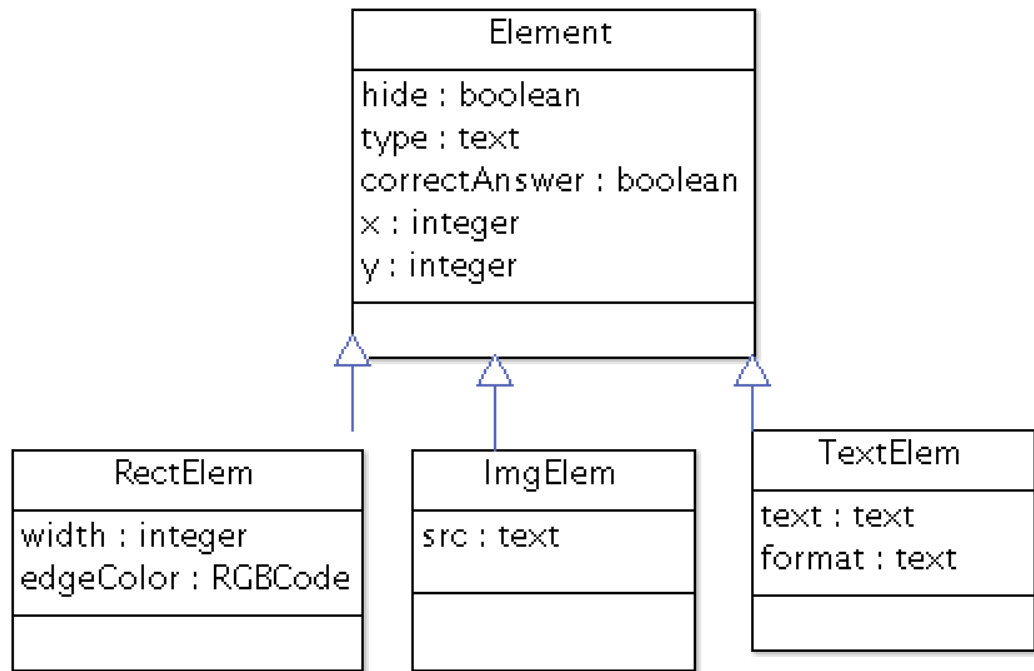


Figura 3-6 –Esquema de los elementos que pueden componer un canvas

Existe un modelo básico (*Element*) con las propiedades comunes de un elemento y del que heredarán el resto de elementos *ImgElem*, *RectElem*, *LineElem*, *TextElem*(Imágenes, rectángulos, líneas y textos).

También hemos redefinido el canvas de HTML como *gameCanvas* y se definirá como un objeto único en el contexto de ejecución que contendrá Elementos extendiendo la funcionalidad estándar del canvas de HTML5.

El *gameCanvas* tiene funciones como controlador y modelo ya que como modelo define el tamaño de la zona de juego y como controlador permite acceso a los elementos ya que es el que mantiene registro de los elementos existentes , se encarga de pintar los elementos y de dar acceso a ellos. El *gameCanvas* sirve por tanto como puente entre el modelo y controlador.

La aplicación almacena la información de los juegos siguiendo el formato JSON y usando *IndexedDB* para guardar los objetos JSON.

Tendremos un objeto array con los tipos de juegos existentes y crearemos un segundo objeto de almacenamiento con los juegos

La información que tienen los objetos de juegos se compone de:

- Nombre
- Titulo
- Tamaño de Canvas
- Elementos del canvas
- Identificador del tipo de juego.

Gracias a un índice sobre el identificador del tipo podremos obtener para cada plantilla los juegos disponibles. Y gracias al array que almacena los identificadores para cada tipo de juego podremos tener control sobre las plantillas existentes que están almacenando juegos. Si quisiéramos añadir nuevas plantillas solo sería necesario registrar un nuevo identificador con nuestra plantilla y contrastar con el array para saber si dicho identificador está en uso.

Gracias a que la información está guardada en formato JSON podremos procesarla, importarla y exportarla con facilidad haciendo uso de ficheros que nos permitirá compartir los juegos con otras personas y usarlos en otros ordenadores.

Como clave única para los juegos se usa la concatenación *TipoNombre* del juego lo que nos permitirá obtener inequívocamente la información de ese juego.

3.5.4 El Controlador

El controlador se compone de una serie de módulos en *JavaScript* que permitirán controlar los eventos que se produzcan en el canvas y modificar el Modelo y en particular los elementos del canvas en función de lo que se establezca hacer en las funciones asociadas al evento. El sistema funciona de la siguiente manera.

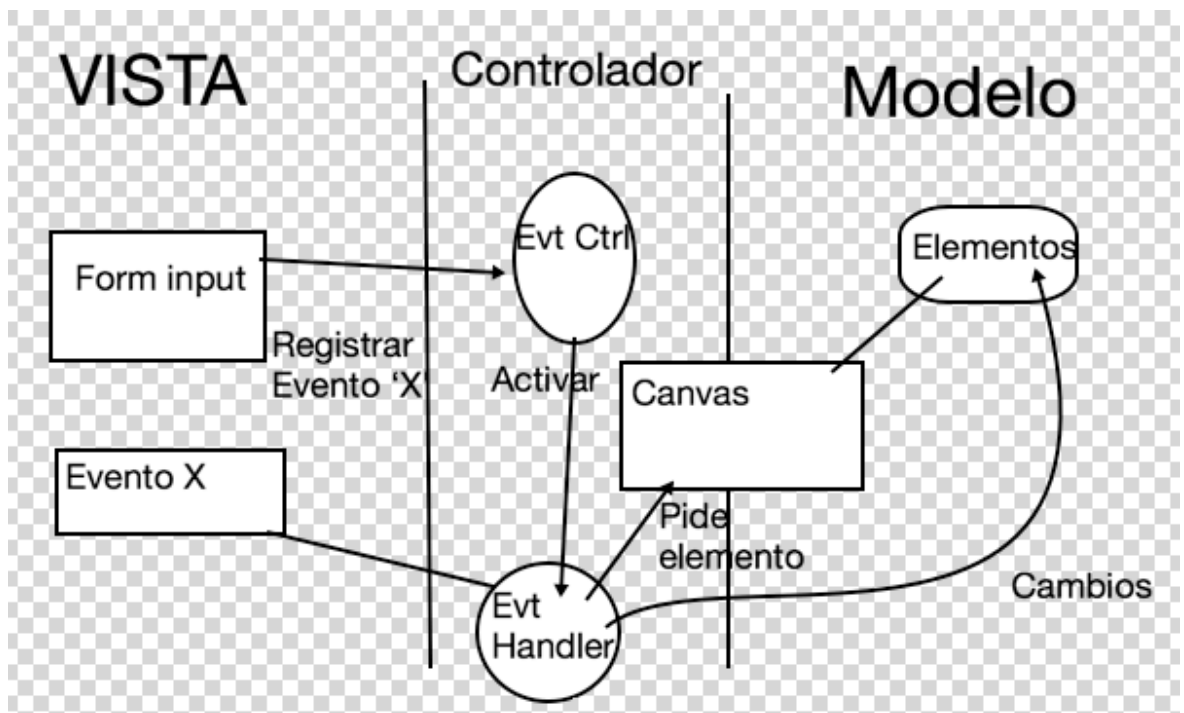


Figura 3-7 Esquema de la interacción entre los componentes de la aplicación

Existe una módulo que mantiene un registro de los eventos para garantizar que solo haya una función encargada de hacer una acción para cada tipo de evento e impedir que se ejecuten múltiples funciones para un mismo evento con los problemas que ello supone. Cuando se quiera realizar una acción sobre el canvas se usará algún elemento de entrada de HTML para registrar un evento en controlador que asignará a una función el control de dicho evento, como por ejemplo asignar el 'click' del ratón a eliminar un elemento. Cuando se produce un evento se activa la función que realiza la acción solicitada. En nuestro ejemplo se preguntaría que elemento se encuentra bajo el ratón, para después eliminarlo por la pulsación del click del ratón.

3.5.5 Las plantillas

La herramienta se basa en plantillas y ejemplos que los profesores pueden modificar. Las plantillas se han hecho siguiendo las referencias de la pedagoga y otros profesores del centro donde se realizaron las pruebas. Esas referencias proceden en varios casos de libros de juegos para niños de 3 a 5 años de la serie Estimular y aprender de Jesus Jarque García [4].

Se han diseñado algunas plantillas que permitan generar juegos como los de dichos libros así como otras que pudieran ser interesantes a partir de las ideas que han surgido con profesores.

Las plantillas y juegos se separan en función del tipo de acción necesario para realizar el juego dependiendo de si es necesario por ejemplo escribir un número, señalar elementos, unirlos con líneas u otro tipo de acciones.

4 Desarrollo

4.1 Introducción

En este apartado vamos a detallar el estado de la aplicación y los detalles de cómo se ha producido desarrollo de cada parte los problemas asociados que ha habido y se aclararán otros puntos de la aplicación como las plantillas que particularmente se han implementado.

4.2 Estructura de archivos del proyecto

La composición de carpetas de la aplicación es la siguiente:

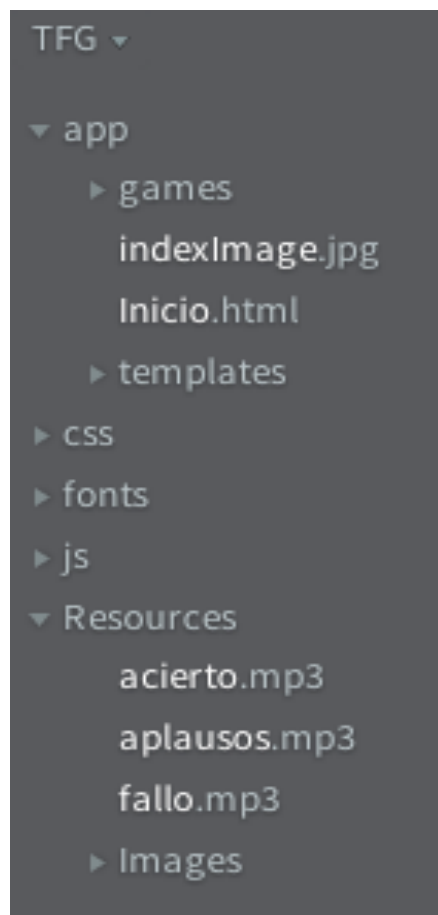


Figura 4-1 Esquema del directorio de la aplicación

En la carpeta *app* están la página de inicio de la aplicación y dos carpetas *templates* y *games* con las páginas de plantillas y juegos respectivamente. En las carpetas *css* y *fonts* se encuentran los archivos que definen el formato de los distintos componentes HTML usados. En *js* encontramos los scripts de los controladores y librerías de apoyo usados en la aplicación. Por último en *Resources* encontramos los

recursos de audio y las imágenes para generar juegos en la herramienta.

4.3 Vista

La vista se ha desarrollado con páginas HTML partiendo del diseño establecido en el apartado 3.

Para el diseño de algunas partes de las páginas se ha usado elementos y funcionalidades facilitados por Bootstrap[5] y dw_tooltips[6]

El primero es un *framework* para desarrollar interfaces de usuario de una manera sencilla, y con un diseño unificado. Facilita una hoja de estilos CSS, junto a extensiones *JavaScript* para desarrollar dichas interfaces. Proporciona plantillas de tipografías, tablas, formularios, botones... es decir, nos ofrece plantillas con las que incluir la mayoría de elementos necesarios para la creación del front-end de nuestras aplicaciones.

Este *framework* fue desarrollado por *Twitter* para unificar hojas de estilos. En estos momentos, es uno de los proyectos más importantes en *Github*, lo que supone que es un *framework* que aporta muchas ventajas para los desarrolladores y que está en continuo crecimiento y mantenimiento, lo que nos ofrece muchas garantías para utilizarlo.

El segundo componente usado es un modulo desarrollado por *Dynamic Web Coding* que implementa una manera sencilla de añadir mensajes de ayuda sobre distintos elementos de HTML y reaccionando ante distintos eventos permitiendo ajustar fácilmente el formato y las características de estos mensajes en el estilo y el comportamiento.

Página de inicio:

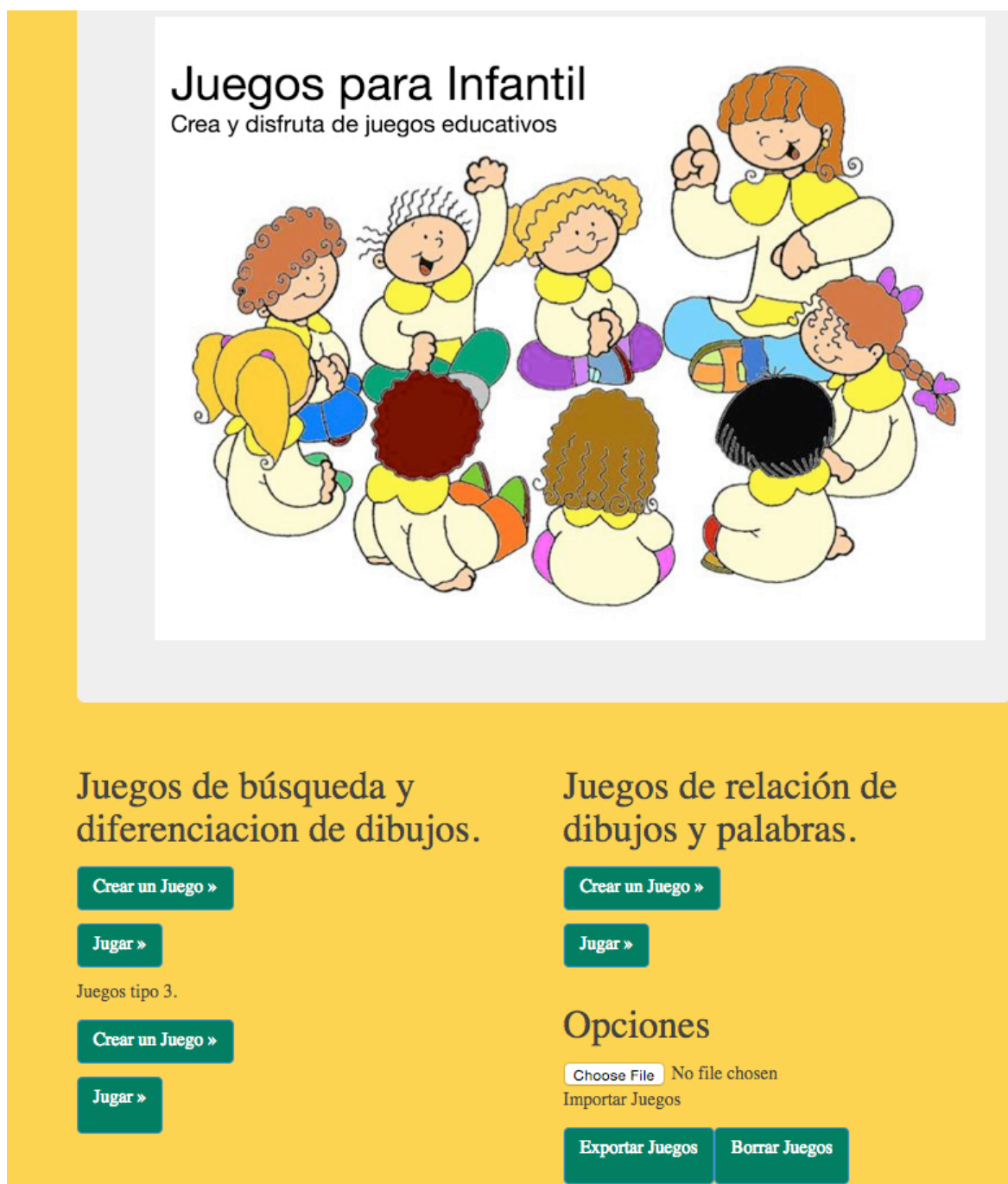


Figura 4-2 Pagina de inicio de la aplicación

La página de inicio mantiene en líneas generales el diseño planteado inicialmente pero se ha decidido añadir las opciones para cargar , exportar y borrar los juegos en esta página.

Plantillas:

Son los ficheros HTML que se encuentran en la carpeta *templates*

The screenshot displays a web application interface for managing a game template. At the top, a green header bar contains the word "Inicio". Below this, a yellow navigation bar features three main sections: "Guardar juego" (Save game) with a "Salvar" button, "Cuadro de juego" (Game board) with input fields for "Ancho" (Width) set to 600 px and "Alto" (Height) set to 400 px, and "Título del juego" (Game title) with a "Cambiar título" button. The central area is a large white canvas, currently empty, with a status bar at the bottom left showing "210,-134" and a "Cargar" button. To the right of the canvas, a yellow sidebar contains a section titled "Establecer numero de relaciones" (Set number of relationships) with an "Insertar" button. Below this, there are options to "Cambiar tamaño de elementos" (Change element size) with width and height inputs, and radio buttons for "Relacionar dos elementos" (Relate two elements), "Mover Elementos" (Move elements), and "Eliminar elemento" (Delete element). Further down, there are links for "Añadir imagen" (Add image) with a "Choose File" button, "Añadir rectangulo" (Add rectangle) with a "+" button, and "Añadir Texto" (Add text) with a "+" button. At the bottom of the sidebar, there are radio buttons for "Editar Texto" (Edit text) with input fields for "Texto" (Text) and "Tamaño" (Size).

Figura 4-3 Plantilla de la aplicación con sus diferentes acciones.

En el caso de las plantillas tenemos una cabecera con tres opciones, guardar juego, cuadro de juego (que permite cambiar el tamaño del canvas) y Título de juego donde se cambia el título de juego.

A la izquierda tenemos el canvas y a la derecha las opciones específicas de las plantillas.

La interacción para cambiar opciones se ha realizado usando elementos de formulario HTML y algunos complementos que nos proporcionaba Bootstrap como los símbolos '+' para añadir algunos elementos al canvas.

A estos elementos de formulario se le han asociado eventos de *JavaScript* que envían al controlador las indicaciones necesarias para que se pueda realizar cambios en el canvas según como indica la opción escogida (Añadir Imagen, añadir textos, imágenes etc).

Por otra parte tenemos los iconos de información que al pulsarse muestran una explicación de cómo usar dicha opción.

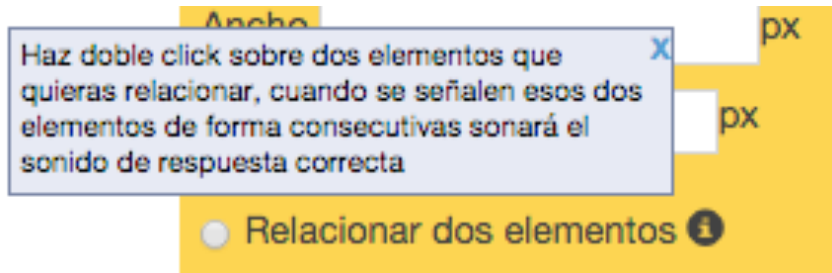


Figura 4-4 Sistema de ayudas

Como se mencionó esto es posible gracias al modulo `dw_tooltip` que nos permite generar fácilmente estas ayudas y configurarlas.

Se insertan dos scripts en la página,

```
<script type="text/JavaScript" src="../../js/dw_tooltip_c.js" ></script>
```

```
<script type="text/JavaScript" src="../../js/helpTexts.js" ></script>
```

El primero implementa la funcionalidad de los carteles de ayuda y en el segundo tenemos definidos los mensajes de ayuda y la configuración

```
dw_Tooltip.defaultProps = {  
  
    supportTouch: true, //compatibilidad con pantallas tactiles  
  
    activateOnClick: true, //compatibilidad con click del raton  
  
    showCloseBox: true, //mostrar icono para cerrar dialogo  
  
    Above: true  
  
}
```

Con el código anterior definimos las propiedades del sistema de ayudas haciendo que las ayudas se activen al pulsar con el ratón, al tocar con el dedo y mostrando un icono para cerrar el dialogo

A continuación tenemos los mensajes de ayuda en un listado donde definimos una variable para cada mensaje para que después podamos referenciarla en las propiedades de la etiqueta correspondiente.

```
dw_Tooltip.content_vars = {  
  
    setTitleHelp: 'Introduce un título o instruccion para el juego',  
    saveGameHelp: 'Introduce un nombre para el juego',  
    .....  
}
```

En custom.css tenemos la configuración visual del cuadro de texto.

```
div#tipDiv {  
    font-size: 11px; line-height: 1.2;  
    color: #000; background-color: #E1E5F1;  
    border: 1px solid #667295; padding: 4px;  
    width: 250px;  
}
```

Con todo esto se generan clases css que podemos usar en los elementos HTML y en particular los iconos de información para que funcionen como iconos que muestren ayudas.

```
<span class="glyphicon glyphicon-info-sign showTip  
    variabaleDelMensajeaMostrar"></span>
```

Paginas de Juego:

Son las páginas HTML que se encuentran en la carpeta games y tienen el siguiente formato.

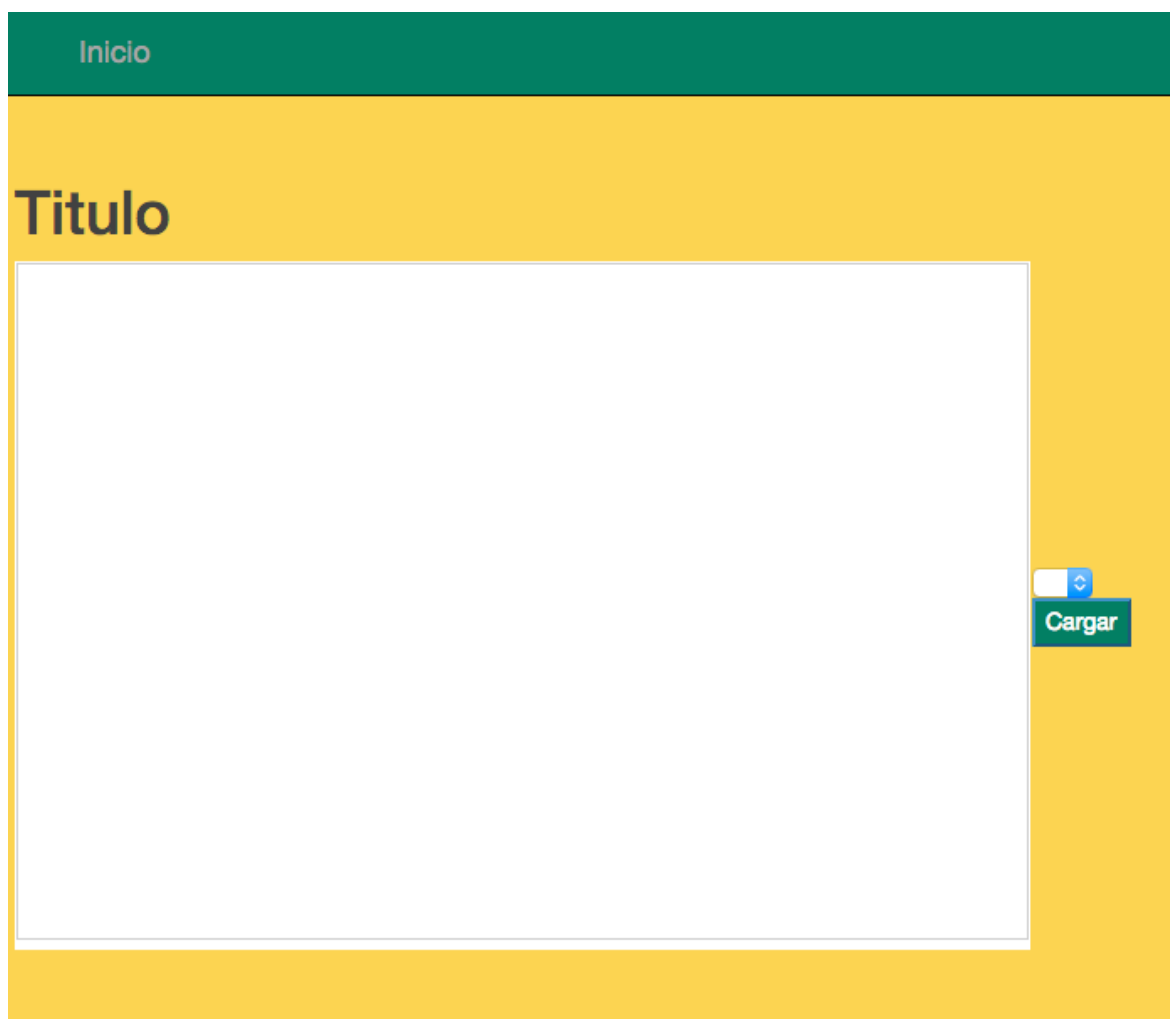


Figura 4-5 Página de juego

Un título de juego que típicamente es la instrucción de cómo completar el juego, un desplegable con los juegos disponibles y el botón de carga.

En cada uno de estos HTML existe una función de *JavaScript* que evalúa en cada momento si se dan las condiciones de éxito para el tipo de juego que corresponde a esa página.

4.4 El modelo

Durante el desarrollo se ha ido mejorando y descubriendo nuevas características necesarias para los elementos del canvas terminando un con un modelo similar al establecido en el diseño pero con propiedades más completas para satisfacer todas las posibilidades que se pudieran dar.

Todos los elementos heredan del base Element y están implementados en el fichero elements.js

```
function Element(x, y){  
  this.x = parseInt(x);  
  this.y = parseInt(y);  
  
  elemIDS++;  
  this.id = elemIDS;  
  this.correct = null; //indica si es una respuesta correcta  
  this.edge= false; //indica si hay que pintar un borde o no  
  this.edgeColor = 'black'; indica el color del borde  
  this.hide = false; //oculto o no  
  this.movable = true; //si se puede mover el elemento  
  this.relations = [];  
  this.type = "elem"; //identificador del tipo de elemento  
}
```

Id es el identificador del elemento. Para garantizar que no se repite hacemos uso de una variable global elemIDS que se incrementa cada vez que se genera un nuevo elemento.

Relations es un array donde se almacenan los identificadores de otros elementos con los que esta relacionado lo cuales se utilizan a la hora de condicionar las reglas de algunos juegos en función de las relaciones como el clásico de relaciones de imágenes

El resto de propiedades no tienen mayor funcionalidad que la indicada en el código marcan aportan utilidades varias a lo hora de generar juegos en la plantilla y dichas propiedades serán usadas por el canvas para, a la hora de dibujar los elementos, tomar decisiones de dibujo asi como para que el controlador pueda tomar decisiones de la lógica del juego en función de los valores de estas propiedades.

Para guardar los juegos se ha creado un objeto *JavaScript gamedb* definido en el archivo gamedb.js que nos servirá de interfaz con la base de datos del navegador implementando las siguientes funciones.

opendb()

Esta función se ejecuta al arrancar cualquier página de la aplicación para garantizar que existe la base de datos y el índice para el objeto que almacena los juegos.

```
var openRequest = indexedDB.open("CanvasGames",1);

    openRequest.onsuccess = function(e) {
        console.log("running onsuccess");
        //db = e.target.result;

    }

    openRequest.onerror = function(e) {
        console.log("Error opening db");
    }

    openRequest.onupgradeneeded = function(e) {
        var thisDB = e.target.result;

        if(!thisDB.objectStoreNames.contains("games")) {
            thisDB.createObjectStore("games").createIndex("type", "type",
{unique:false});
        }
    }

},
```

Debido a que *indexedDB* usa un sistema de llamadas asíncrono es necesario definir una función de éxito y de error cada vez que queremos realizar una acción sobre la base de datos.

La sentencia *indexedDB.open* recibe un parámetro y un identificador para la base de datos. Después se definen las llamadas funciones para *onsuccess* y *onerror* que serán llamadas en función del resultado de la operación.

Al usarse como se ha dicho un sistema de llamadas asíncrono se realizará una llamada a *open* y se definirán *onsuccess* y *onerror* cada vez que se quieran introducir y obtener datos.

- *addNewType* (id, callback)

Con esta función se registra un nuevo tipo de juego en la base de datos

- *addGame*((id, name, title, canvas)

Con esta función añadimos un juego a la base de datos.

Para ello una vez abierta la conexión asignamos la siguiente función a *onsuccess*:

```
Function (e){  
  
    db = e.target.result;//Obtenemos la conexión  
    //comenzamos una transacción para el objeto de juegos  
    var transaction = db.transaction(["games"],"readwrite");  
  
    //usando la transacción obtenemos el almacén de juegos.  
    var store = transaction.objectStore("games");  
    //Añadimos el juego  
    var request = store.add(game,game.type.concat(game.name));  
  
    request.onerror = function(e) {  
        console.log("Error",e.target.error.name);  
        //some type of error handler  
    }  
    request.onsuccess = function(e) {  
        console.log("Exito");  
    }  
}
```

En primer lugar se crea una transacción para realizar la operación, en segundo lugar se obtiene el objeto de almacenamiento y por último se introduce el nuevo juego con la función *add*.

-getGamesForType(id, callback);

Devuelve los juegos para un tipo dado. Al igual que la función anterior crea una transacción y solicita acceso al objeto de almacenamiento para finalmente obtener el índice de tipos con

```
var index = store.index("type");
```

y asignar a la función *onsuccess* la función pasada por parámetro y que hará uso de los resultados de la petición.

```
var request = index.openCursor(id).onsuccess=callback;
```

-getGame(id, name, callback)

Su funcionamiento es equivalente al anterior con la diferencia de que hace uso del identificador del tipo de juego y el nombre para recuperar el juego en particular:

```
var request = store.get(id.concat(name));  
request.onsuccess = callback;
```

El resultado se recupera en la función del *callback*.

- eraseGamesOfType(id)*
- eraseGame(key)*
- eraseAll()*

Estas tres funciones son las encargadas de borrar la base de datos. La primera borra todos los juegos de un tipo, la segunda un juego por clave y la tercera borra todos los juegos haciendo uso de las dos primeras. Para ello un bucle sobre el objeto *Types* con los tipos realiza la llamada a *eraseGamesOfType* para cada tipo. Después se compone la clave concatenando cada nombre con el identificador de tipo y llamando a *eraseGame*.

4.5 El controlador

Las funcionalidad desempeñada por el controlador se encuentra distribuida en los archivos *game.js*, *dbController.js* y *commonAction.js* así como funciones específicas para cada plantilla y juego en los correspondientes HTML.

En *game.js* encontramos *gameCanvas* que se encarga de ampliar las funcionalidades del canvas original de HTML 5 y sirve para controlar y cambiar los elementos dentro del canvas.

Se compone de:

- Canvas*: Donde se almacena el objetos canvas original de HTML5
- mposx*: la posición en el eje x del ratón respecto al canvas.
- mposy*: la posición en el eje y del ratón respecto al canvas.
- Elements*: El array donde se guardan los elementos del canvas.

Al iniciar el objeto canvas se asocia el evento del movimiento del ratón para mantener el control del ratón sobre el canvas actualizando siempre los valores *mposx* y *mposy*.

Gracias a este control es posible solicitar el elemento que se encuentra bajo el ratón cada momento a través de la función *getElementUnderMouse*.

Después se implementan las funciones que permiten modificar los elementos.

- addElement(Elem)*: Sirve para añadir un elemento.
- getElementBy(id)*: Para obtener un elemento en función del id
- getElementByType(type)*: Para obtener un array con los elementos de un tipo particular.
- removeElement(elem)* : Para remover un elemento del canvas.

Otro métodos de ejemplos para modificar las propiedades los elementos son:

- showEdgesForType(boolean, type)*: Sirve para decidir se se muestra el bordeado de los elementos del canvas para un tipo dado

-hideElementsOfType(boolean, type): Sirve para decidir si se muestra o no los elementos del canvas para un tipo dado en *type*.

-hideElement(boolean, elem): Ocultar un elemento o concreto o dejar que se muestre.

Existen más funciones con funcionalidad similar a las anteriores pero por no entrar en un listado demasiado largo y dado que la lógica que implementan es esencialmente es la misma (podrían compararse a *getters* y *setters* típica forma de acceso a propiedades de un objeto en la programación orientada a objetos), no se ha querido alargar este apartado.

También es posible modificar directamente las propiedades de un elemento después de obtenerlo con llamadas al *gameCanvas*

En *game.js* define el objeto *gameEvents* que se encarga de gestionar los eventos que se asocian al canvas.

```
//Gestor de eventos
var gameEvents = {

    setEvent: function(type, handler){
        var id = "#" + gameCanvas.canvas.id
        $(id).unbind(type);
        $(id).on(type, handler);
    }

}
```

El motivo de este objeto era controlar que no se repitiese ningún evento sobre el canvas para más de una función ya que no es posible saber el número de funciones asociadas a un mismo evento en *JavaScript*. Sin embargo se descubrió que la librería *JavaScript jQuery*[6] altamente extendida para manipular elementos de HTML y que permitie también asociar eventos dispone de una función *.unbind* que permite eliminar todas las asociaciones entre un evento y las funciones.

De esta manera solo es necesario registrar los eventos usando la función *setEvent*. Esta limpia primero las asociaciones existentes con *.unbind* para después registrar el evento y función pasados por parámetro (*type* y *handler*) usando la función *on* en el canvas (el identificador de este elemento lo obtenemos de *gameCanvas*). De esta forma garantizamos que solo existe un evento para cada función

En *commonnActions.js* están implementados un conjunto de funciones útiles para asociar a los eventos y realizar acciones sobre el canvas.

El siguiente siguiente ejemplo sirve para remover un elemento del canvas.

```
//Habilitamos el doble clic del raton para la funcion removeElem  
function enableRemoveElements(event){
```

```
    gameEvents.setEvent(event, removeElem);  
}
```

```
//Funcion que elimina el elemento del canvas
```

```
function removeElem() {  
    var elem = gameCanvas.getElementUnderMouse();  
    gameCanvas.removeElement(elem);  
    for (key in elem.relations) {  
        var elem2 = gameCanvas.getElement(key);  
        gameCanvas.removeRelation(elem, elem2)  
    }  
    gameCanvas.draw();  
}
```

```
<input type="radio" name="action" onclick="enableRemoveElements('dblclick')">
```

Cada vez que se registra un evento que queremos que se detecta sobre el canvas se hace uso de algún elemento HTML en la plantilla y en la etiqueta se llama a la función que habilita la acción.

En este caso la acción sería registrada en el *gameEvent* como se ve en la función *enableRemoveElements*. De esta forma garantizamos que solo la acción *removeElement* es ejecutada cuando se produce un doble click.

En la función registrada se ejecuta la lógica necesaria, en este caso se solicita el elemento y gracias a las funciones del canvas se elimina el elemento usando *gameCanvas.removeElement*.

Esta forma de establecer una acción es la que se aplica en todos los casos. Cada vez que se quiere enlazar un evento y función con el canvas se hace uso de una etiqueta de formulario u otras etiquetas HTML compatibles con los eventos asociados a estas (onclick, onchange...), para registrar un evento en *gameEvents*.

Muchos de estos eventos y acciones están implementados en *commonActions.js* al igual que el ejemplo anterior, pero si se necesita alguna acción más específica se puede definir un nuevo fichero *JavaScript* o simplemente añadir la funcionalidad concreta dentro de la propia página de la plantilla o juego. En este último caso es lo habitual ya que la función que se asocia a un evento para analizar el estado y comprobar si se ha superado un juego es única para cada uno de los tipos de juego.

Por último en *dbController.js* se implementan las funciones que se usan para obtener los datos, el listado de juegos de un tipo o generar e importar ficheros sirviendo de coordinación entre la vista y el modelo.

En el se implementan las funciones:

- save*: Para guardar un juego en las páginas de las plantilla
- insertGames*: Para guardar un array de juegos típicamente en el caso de la importación.
- loadGamesOfType*: Para cargar el listado de juegos en una lista HTML
- loadGame*: Que carga un juego específico de la base de datos .

4.6 Las Plantillas

Las plantillas se han diseñado en función del objetivo del juego que se podían deducir de las conversaciones con profesores y los ejemplos vistos en cuadernos de ejercicios.

Los objetivos abstraídos han sido los siguientes

- 1-Señalar uno o varios elementos.
- 2- Unir dos elementos con una línea
- 3- Mover elementos a una posición
- 4- Introducir números típicamente para pruebas matemáticas.
- 5- Introducir una palabra

Teniendo en cuenta estos cinco objetivos se han diseñado cinco plantillas donde los juegos serán evaluados en función de los 5 objetivos descritos arriba. También delimitará las opciones disponibles en cada plantilla y junto a los ejemplos los profesores tendrán una línea a seguir para la creación de juegos .

En el anexo de este trabajo se encuentra el manual de usuario donde se podrá ver con algo de más detalle las plantillas y las propiedades de cada una.

4.7 La aplicación para tablets

Debido a que los sistemas operativos móviles (IOS y Android) no dan acceso al sistema de ficheros no es posible ejecutar la aplicación como lo haríamos en un sistema operativo de escritorio, por ello se ha creado una versión paralela de la aplicación que incorpora un modulo en php para permitir subir ficheros de juegos y cargarlos desde una nueva página no disponible en la versión local. De esta manera se consigue que se pueda jugar desde *tablet* conectándose desde el navegador.

5 Integración, pruebas y resultados

5.1 Introducción

En este apartado detallaremos en que han consistido las pruebas que se han realizado por un lado para corroborar la integración , y por otro las pruebas de usabilidad con alumnos y profesores y expondremos las encuestas realizadas.

5.2 Pruebas de Integración

Se ha diseñado la siguiente prueba para comprobar que el sistema es compatible en múltiples sistemas y *navegadores*.

Transferir en un pendrive la carpeta de la aplicación tres maquinas con los sistemas operativos de escritorio más utilizados hoy en día OSX, Ubuntu y Windows , ejecutar la aplicación y se realizar las siguientes tareas:

- Crear un juego
- Ejecutarlo
- Superar el juego
- Exportar el juego
- Borrar la base de datos
- Importar el juego exportado anteriormente.
- Cargar el juego

Para comprobar que se podían usar en *tablet* se traspasó la aplicación a un servidor y se precargaron unos juegos.

Después se ha establecido una conexión con una *tablet* nexus 7 y un Iphone 4s para probar la compatibilidad con safari para IOS y chrome para android. Para ellos se ha cargado un juego de los disponibles en el apartado especial de la versión para servidor de la aplicación y se accedido a ella con dichos dispositivos

5.3 Resultados de la prueba de integración

Sistema\Prueba	Crear juego	Ejectar Juego	Superar Juego	Exportar Juego	Importar Juego	Cargar Juego
OSX - Safari	✓	✓	✓	✓	✓	✓
OSX - Firefox	✓	✓	✓	✓	✓	✓
OSX - Chrome	✓	✓	✓	✓	✓	✓
Windows-Chrome	✓	✓	✓	✓	✓	✓
Windows-FireFox	✓	✓	✓	✓	✓	✓
Ubuntu-Firefox	✓	✓	✓	✓	✓	✓
Ubuntu-Chrome	✓	✓	✓	✓	✓	✓
Chrome -Android		✓	✓			✓
Safari - iOS		✓	✓			✓

Tabla 5-1- Resultados de compatibilidad en navegadores

La compatibilidad entre navegadores de escritorio es amplia como era de esperar dado que se han usado las tecnologías más apropiadas para lograr dicha compatibilidad. No ha sido posible probar las versiones más actuales de Internet Explorer, la única versión probada, la 8 no soportaba la etiqueta de canvas.

En cuanto a la compatibilidad con dispositivos móviles es suficiente para ejecutar los juegos en tablets y móviles

5.4 Pruebas de usabilidad

Se han realizado dos pruebas, una para profesores y otra para alumnos, para comprobar que la herramienta puede usarse para el propósito para el que desarrollado.

Las pruebas han sido desarrolladas en un aula de infantil del centro Mater Inmaculata.

5.4.1 Prueba para los profesores

Se le mostró a la coordinadora y profesora del nivel de educación infantil del centro tres ejemplos de juegos, dos basados en libro de referencia que se ha usado para diseñar las plantillas y otro de cuentas matemáticas. Se le pidió que los reprodujera haciendo uso de la herramienta y usando las ayudas disponibles en la plantilla a través de los botones de información.

De esta manera se ha podido comprobar si es posible realizar el cometido para el que fue desarrollado la herramienta que es ayudar a crear versiones digitales de muchos juegos habituales para el aprendizaje con cuadernos de juegos.

5.4.2 Resultado de la prueba para profesores

Los juegos de referencia y resultados fueron los siguientes:

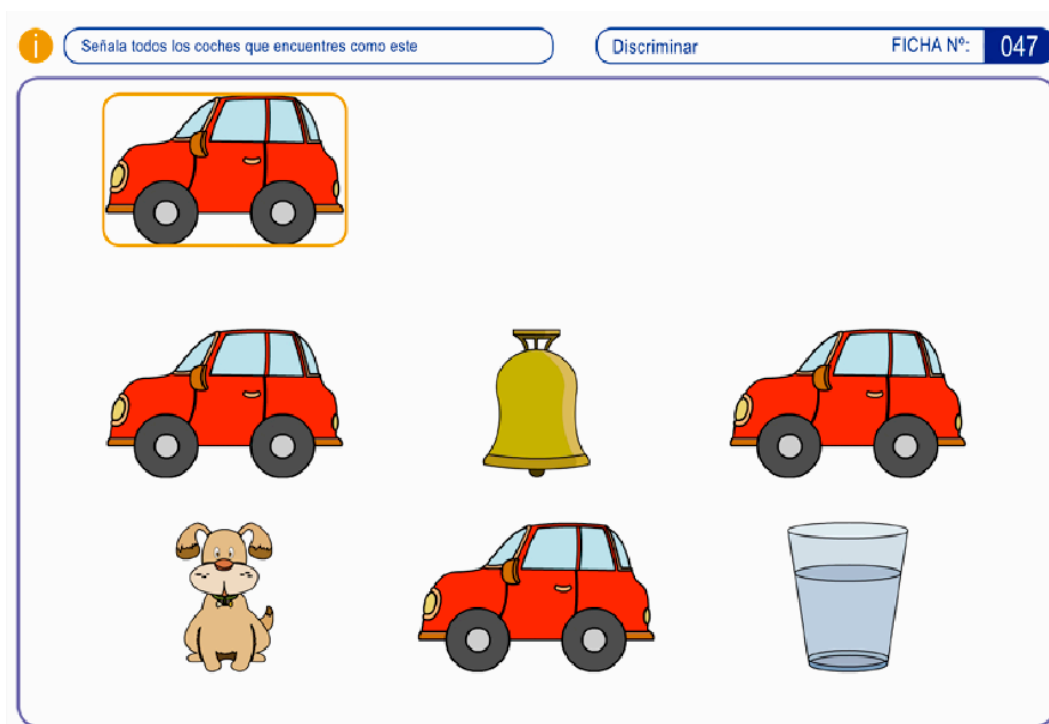


Figura 5-1. Primer Juego de referencia



Figura 5-2 Resultado del primer juego

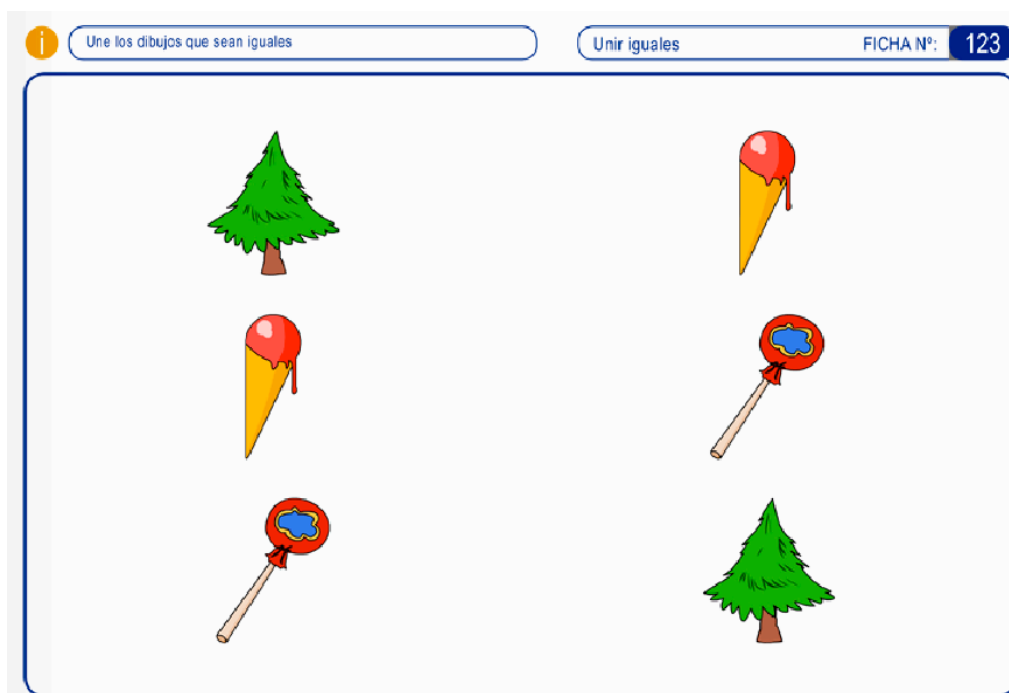


Figura 5-3 Juego de ejemplo relaciones

asociar imagen con palabras con c



Figura 5-4 Resultado del segundo juego creado

sumas

$\begin{array}{r} + 2 \\ 6 \\ \hline \square \end{array}$	$\begin{array}{r} + 6 \\ 3 \\ \hline \square \end{array}$	$\begin{array}{r} + 1 \\ 6 \\ \hline \square \end{array}$	$\begin{array}{r} + 3 \\ 4 \\ \hline \square \end{array}$
<div style="display: flex; justify-content: space-around; font-size: 24px;"> 1 2 3 4 5 6 7 8 9 </div>			

Figura 5-5 Juego de cuentas matemáticas creado en la prueba

Después de la prueba se realizó una breve encuesta sobre la experiencia de la prueba.

Preguntas	Nada/Poco/Bastante/Mucho
¿En tu opinión los recursos han sido...?	Bastantes
¿Te han parecido claros los mensajes de ayuda?	Bastante
¿Te ha parecido clara la distribución de las acciones?	Bastante
¿Te ha resultado cómoda la manera de realizar las acciones?	Bastante
¿En general, te ha resultado fácil la prueba?	Bastante

Tabla 5-1 Preguntas de la encuesta a la profesora

La prueba se realizó sin problemas y la profesora solo mencionó que sería apropiado ordenar la lista de acciones de forma diferente ya que no estaban ordenadas de una manera lógica e intuitiva. Sin embargo afirmó sentirse cómoda realizando las acciones en particular.

Dado que era una profesora que no estaba habituada a manejar ordenadores hasta hace relativamente pocos años es un buen resultado que una persona con ese perfil encontrara las pruebas bastante fáciles.

5.4.3 Pruebas para alumnos

Para comprobar la reacción de los niños con los juegos y su interacción con la pizarra digital se probaron distintos juegos en un aula de 28 niños de 5 años durante aproximadamente 1 hora mientras la profesora observaba.

Se les hicieron tres sencillas preguntas a los niños para conocer si la aplicación había gustado y se hicieron varias preguntas a la profesora para saber como había sido en su opinión la reacción de los niños.

5.4.4 Resultado de las pruebas para alumnos

En la siguiente gráfica podemos ver la respuesta de los niños a las siguientes preguntas:

- 1 - ¿O han gustado los juegos?
- 2 - ¿Jugaríais en casa a estos juegos?
- 3 - ¿Tenéis Tablet u ordenador en casa?

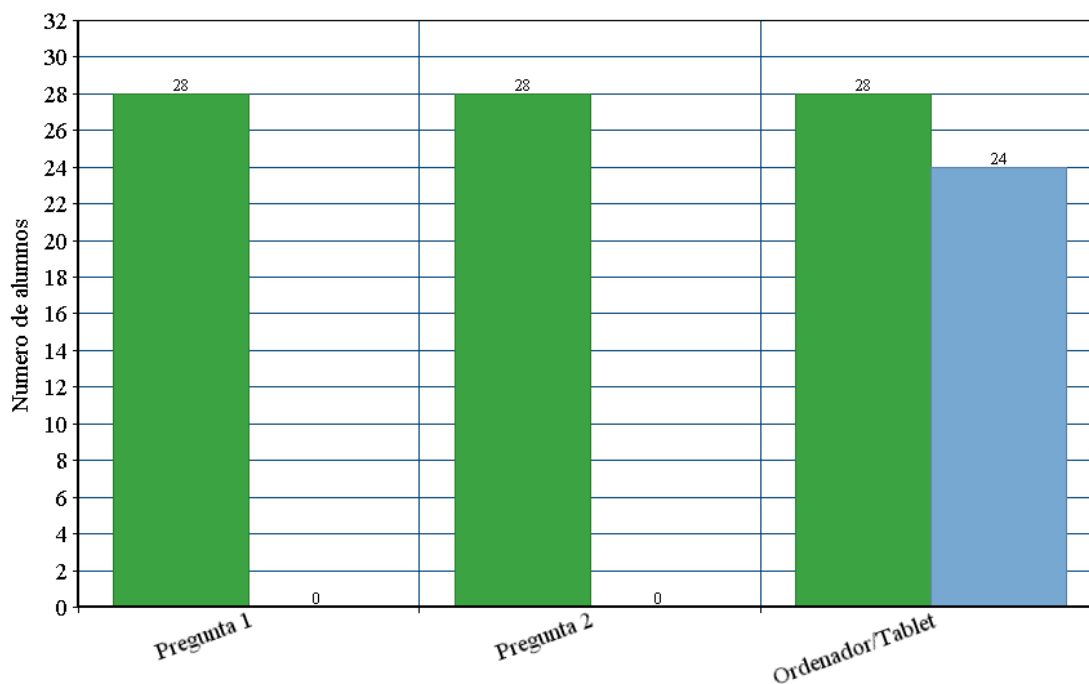


Figura 5-6 – Resultado de la encuesta a niños.

Como muestra la gráfica la aplicación agrado a los niños que parecen sentirse muy cómodos jugando con la pizarra digital.

Y lo que es mucho más interesante 24 de 28 niños tenían tablet y todos tenían ordenador , lo cual refuerza la utilidad de la aplicación para facilitar el aprendizaje en casa usando los mismos juegos que se utilizan en el colegio. En muchos casos los colegios poseen servidores para alojar paginas web y solo sería necesario subir la aplicación a dicho servidor para que pudiesen jugar los niños desde las tablets. Para que jugasen con el ordenador bastaría con mandarles la aplicación por correo o descargarla desde la versión de servidor.

Para comparar como había sido el uso de los juegos en la pizarra respecto a juegos en papel.

Pregunta	Respuesta
¿Te ha parecido lmejor a reacción de los niños usando la pizarra respecto a ejercicios similares en cuaderno?	Si
¿Crees que los niños han reaccionado con más agilidad a las pruebas que con otras similares?	Si
¿Crees que han presatado más atención y han estado más motivados?	Si

Tabla 5-2 Preguntas sobre la reacción de los niños

Como se puede ver la impresión de la profesora fue buena y parece que la línea de desarrollo seguida ha sido exitosa según las pruebas reales.

Los niños se mostraron interesados en participar y hacían lo posible por ser los escogidos para hacer pruebas.

La profesora y coordinadora de infantil ha tenido el detalle además de realizar un corto escrito con sus impresiones que se encuentra en el anexo de este trabajo.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

Después de comparar los resultados con los objetivos y requisitos propuestos se puede considerar que la herramienta cumple con ellos.

Se ha desarrollado una herramienta pensando desde el primer momento en las necesidades del usuario final, en este caso niños y en especial profesores, y que aprovecha al máximo las capacidades de las nuevas tecnologías web de HTML5 y *JavaScript* que prometen ser el futuro desarrollar aplicaciones multiplataforma. También se han exprimido los recursos que otorga el navegador como plataforma de ejecución para lograr un herramienta que no dependa de internet. Todo esto ha supuesto un aprendizaje importante y ha permitido obtener una conocimiento más preciso y completo de lo que es posible hacer con esta tecnología.

Por otra parte las pruebas en un aula real demuestran la necesidad que había en el entorno educativo de una herramienta de este tipo y la utilidad de poder compartir juegos con las mismas características pedagógicas en el aula y en casa. Y lo más importante, los niños han tenido una reacción favorable y les gusta realizar ejercicios como los que se pueden generar con la herramienta.

6.2 Trabajo futuro

Se podría mejorar la integración con servidores ya que se ha diseñado dicha versión para poder jugar en *tablets* y compartir juegos mientras que la creación de nuevos juegos sobre el servidor es limitada ya que se depende de los recursos del cliente y no se han implementado servicios para usar imágenes almacenadas en el servidor, aunque es posible descargar los recursos.

Por otra parte sería deseable investigar las nuevas herramientas de desarrollo para implementar aplicaciones móviles a partir de HTML5 y *JavaScript* para crear aplicaciones nativas para dispositivos móviles.

Referencias

- [1] “Indexed Database API”, W3C Candidate Recommendation 04 July 2013,
<http://www.w3.org/TR/IndexedDB/>
- [2] “JavaScript Web APIS”, Standard,
<http://www.w3.org/standards/webdesign/script>
- [3] <http://www.json.org/>
- [4] <http://www.gesfomediaeducacion.com/4-estimular-y-aprender>.
- [5] <http://getbootstrap.com/>
- [6] <http://www.dyn-web.com/code/tooltips/>

Glosario

API	Application Programming Interface
JSON	JavaScript Object Notation
W3C	World Wide Web Consortium
HTML	Hyper Text Markup Language
MVC	Model View Controller

Anexos

A Manual de Usuario

INSTALACIÓN

Para instalar la aplicación solo tienes que copiar la carpeta con el contenido de la aplicación a cualquier directorio del ordenador.

USO

Abre el archivo inicio.HTML situado en la carpeta App. El navegador predeterminado de tu ordenador se abrirá mostrando la ventana inicial de la aplicación.



Encontrarás 5 tipos de plantillas disponibles y un apartado de opciones.


Las plantillas tienen las siguientes opciones:

El esquema de la plantilla de juego se divide en cinco zonas numeradas:

- 1**: Zona superior izquierda con el botón "Guardar juego" y un campo de texto para el nombre del juego.
- 2**: Zona superior central con el botón "Cuadro de juego" y campos de texto para "Ancho" (600 px) y "Alto" (400 px).
- 3**: Zona superior derecha con el botón "Título del juego" y un campo de texto para el título.
- 4**: Zona inferior derecha con el botón "Establecer numero de relaciones" y una lista de acciones: "Insertar", "Cambiar tamaño de elementos" (con campos de ancho y alto), "Relacionar dos elementos", "Mover Elementos", "Eliminar elemento", "Añadir imagen" (con botón "Choose File"), "Añadir rectángulo", "Añadir Texto" y "Editar Texto" (con campos de texto y tamaño).
- 5**: Zona inferior izquierda con un campo de texto para el nombre del juego y un botón "Cargar".

Figura A-2 Esquema de una plantilla

- 1- **Guardar Juego:** Introduce un nombre y pulsa salvar para guardar el juego.
- 2- **Cuadro de juego:** introduce los valores que quieras para ajustar el tamaño del canvas a tus necesidades
- 3- Introducir instrucción del juego.
- 4- Acciones disponibles en la plantilla
- 5- **Carga de juegos:** Encontrarás un desplegable con los juegos creados y algunos ejemplos que vienen con la aplicación. Selecciona el deseado y pulsa cargar para verlo.

Además tendrás un botón de información  que podrás pinchar para mostrar un cuadro de texto con información útiles sobre que hacer con la acción correspondiente. No dudes en usarlos para guiarte!

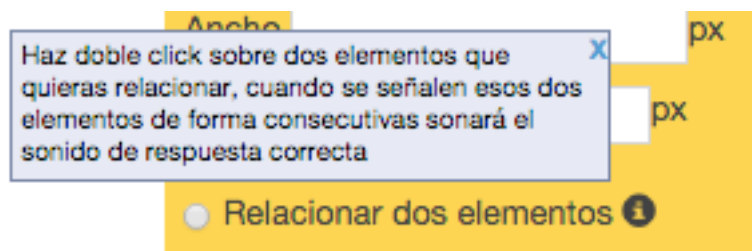


Figura A-3 Ejemplo de sistema de ayudas

Plantilla 1: Juego de búsqueda y diferenciación de dibujos

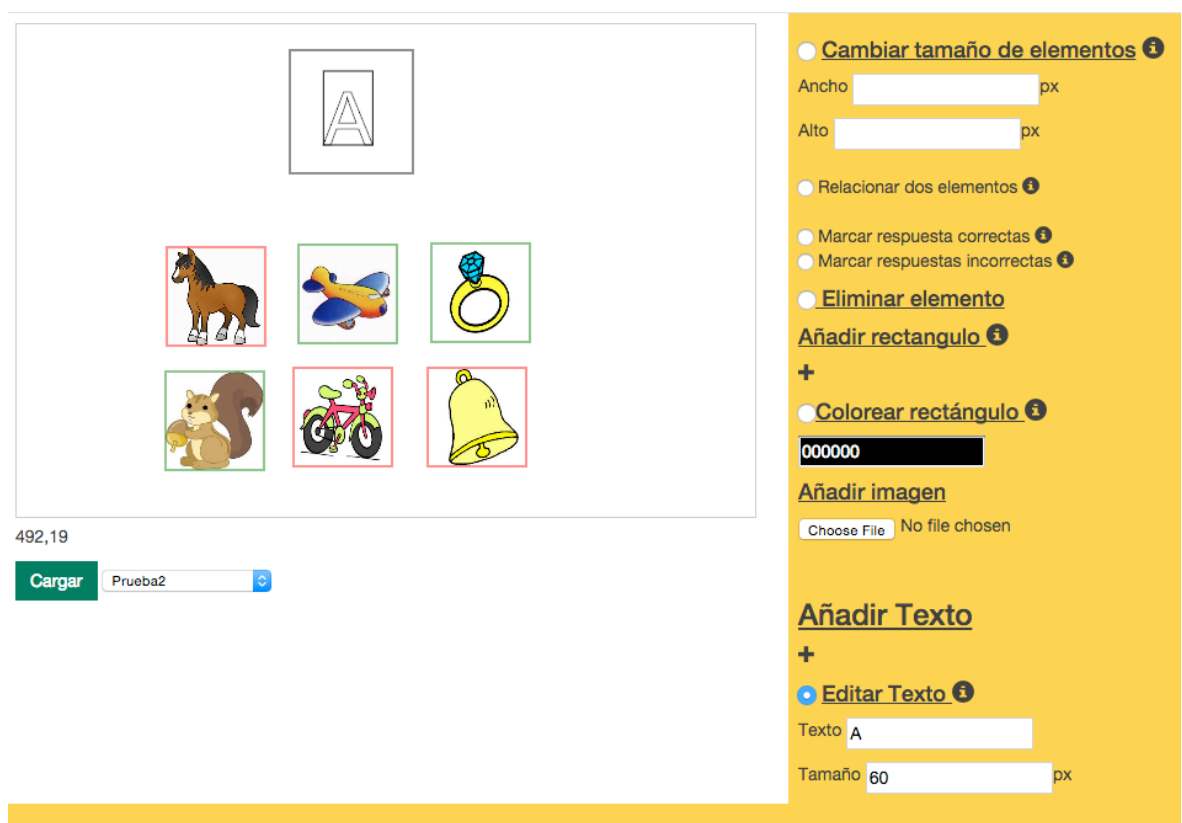


Figura A-4 Plantilla 1 de la aplicación

Esta plantilla sirve para generar juegos cuyo objetivo es señalar imágenes o elementos que cumplan con lo señalado. A la derecha encontraras las opciones que puedes usar para añadir elementos. Usa las acciones *Relacionar dos elementos* , *marcar respuesta correcta* y *marcar respuesta incorrecta*, para indicar que elementos realizarán que sonidos cuando sean pulsados.

Ejemplo

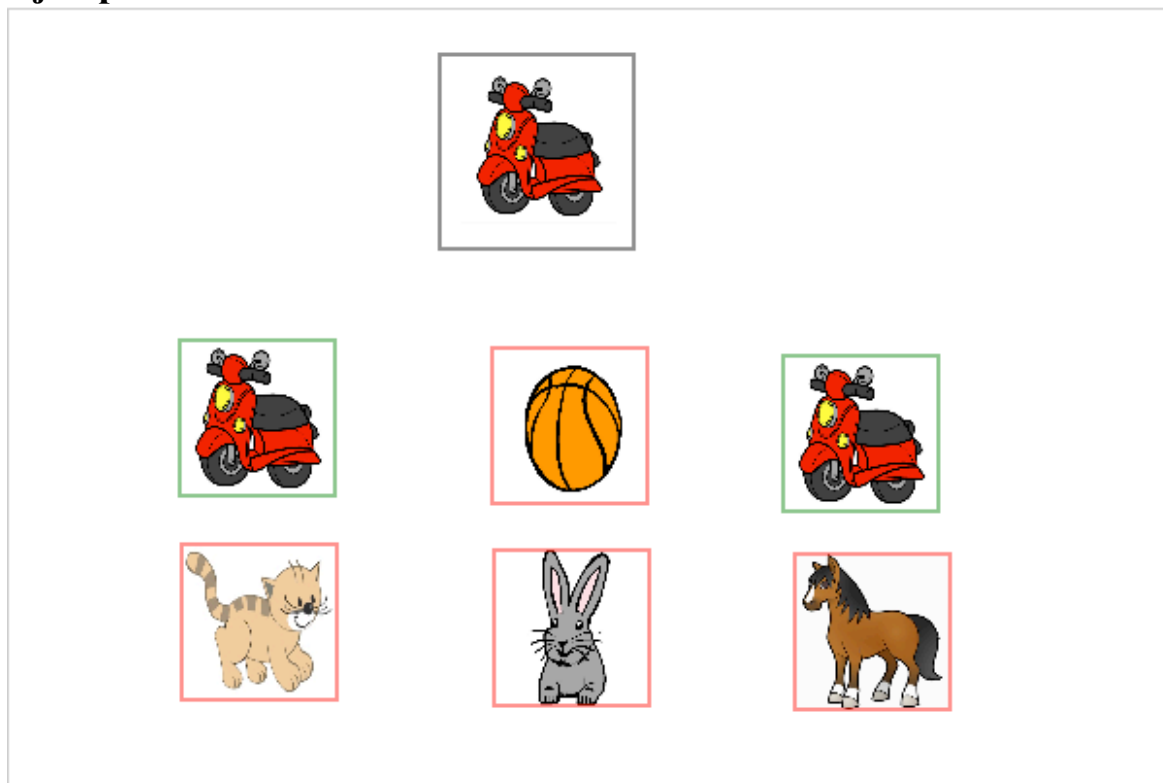


Figura A-5 Ejemplo de juego de plantilla 1

Como puedes ver en el ejemplo las motos son respuestas correctas y por eso aparecen en verde, mientras que el resto de imágenes están marcadas en rojo (respuestas incorrectas).

Cuando cargues el juego la aplicación detectará las respuestas y según lo pulsado sonará un sonido de acierto o fallo.

Plantilla 2: Juegos de relación

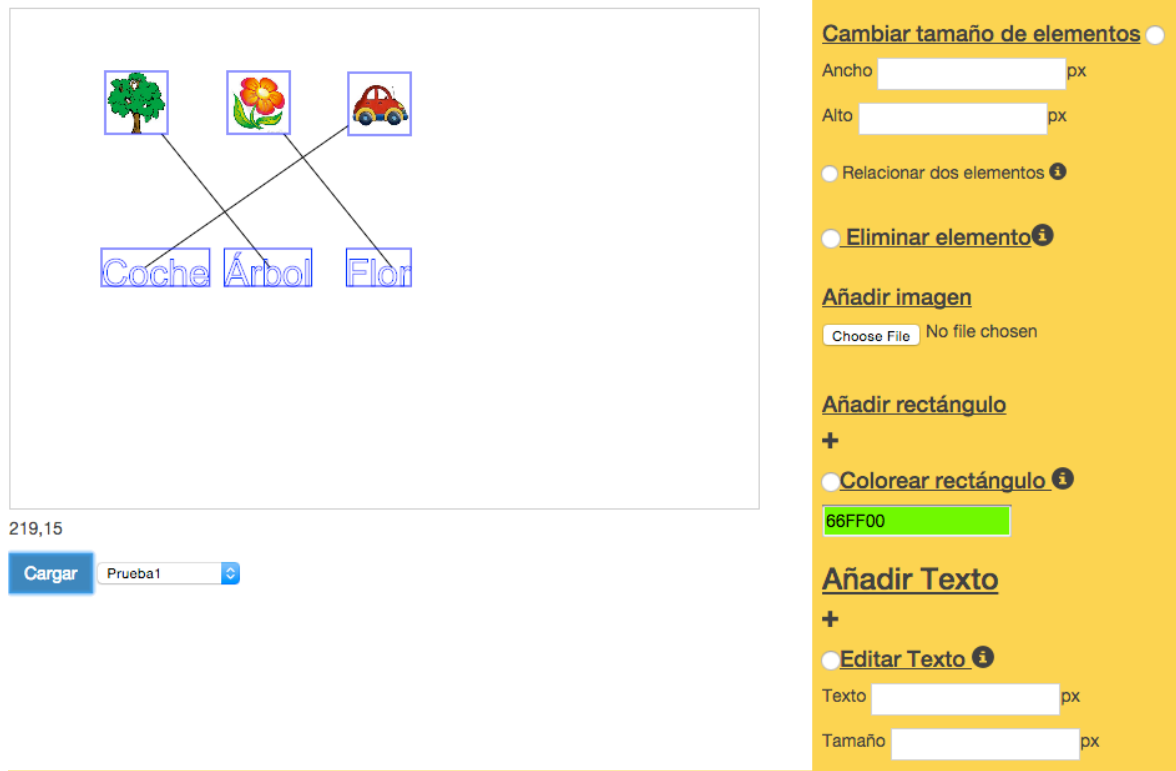


Figura A-6 Plantilla 2 de la aplicación

Esta plantilla sirve para generar juegos cuyo objetivo es unir dos imágenes o elementos que cumplan con la instrucción indicada. A la derecha encontraras las opciones que puedes usar para añadir elementos. Usa las acciones *Relacionar dos elementos* para indicar que elementos hay que unir para que suene el sonido de acierto. Obviamente cuando cargues el juego las indicaciones y uniones que puedes ver al crear el juego en la plantilla desaparecerán para que se puede jugar con normalidad

Ejemplo

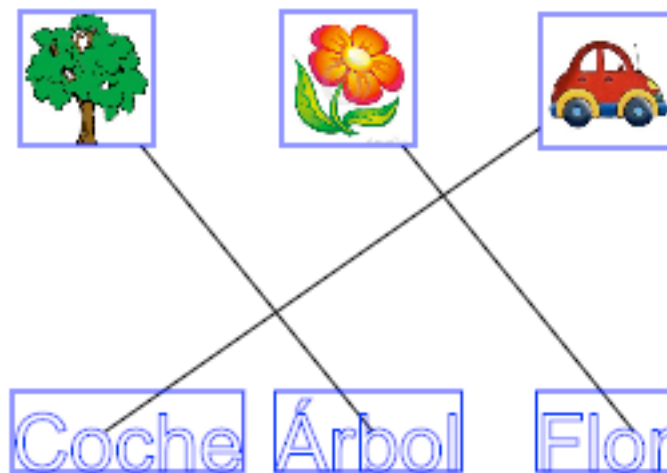


Figura A-7 Ejemplo de juego de plantilla 2

En este ejemplo se han añadido tres imágenes y tres palabras y se han relacionado usando la opción correspondiente.

Plantilla 3: Juegos de operaciones matemáticas.

Número de operaciones ⓘ

Insertar 3

Cambiar tipo de operación ⓘ

☐ +

☐ -

☐ x

☐ Editar Texto ⓘ

Texto px

Tamaño px

103.01388835906982,0.9999847412109375

Cargar

Figura A-8 Plantilla 3 de la aplicación

Esta plantilla permite generar una lista de operaciones cuyos valores y tipos de operación puedes cambiar usando las acciones de la derecha.

Para resolver un operación pulsa el recuadro bajo ella y después pulsa los números necesarios para indicar el resultado.
Finalmente señala otra vez el recuadro para comprobar el resultado.

Ejemplo

$$\begin{array}{r} + 2 \\ 6 \\ \hline 8 \end{array}$$
$$\begin{array}{r} + 6 \\ 3 \\ \hline \end{array}$$
$$\begin{array}{r} + 1 \\ 6 \\ \hline \end{array}$$
$$\begin{array}{r} + 3 \\ 4 \\ \hline \end{array}$$

1 2 3 4 5 6 7 8 9

Figura A-9 Ejemplo de juego de plantilla 3

En este caso tenemos una serie de sumas y la primera ha sido resuelta por ello está marcada en verde. Por otro lado la siguiente cuenta está marcada para ser resuelta.

Plantilla 4: Juegos de mover elementos

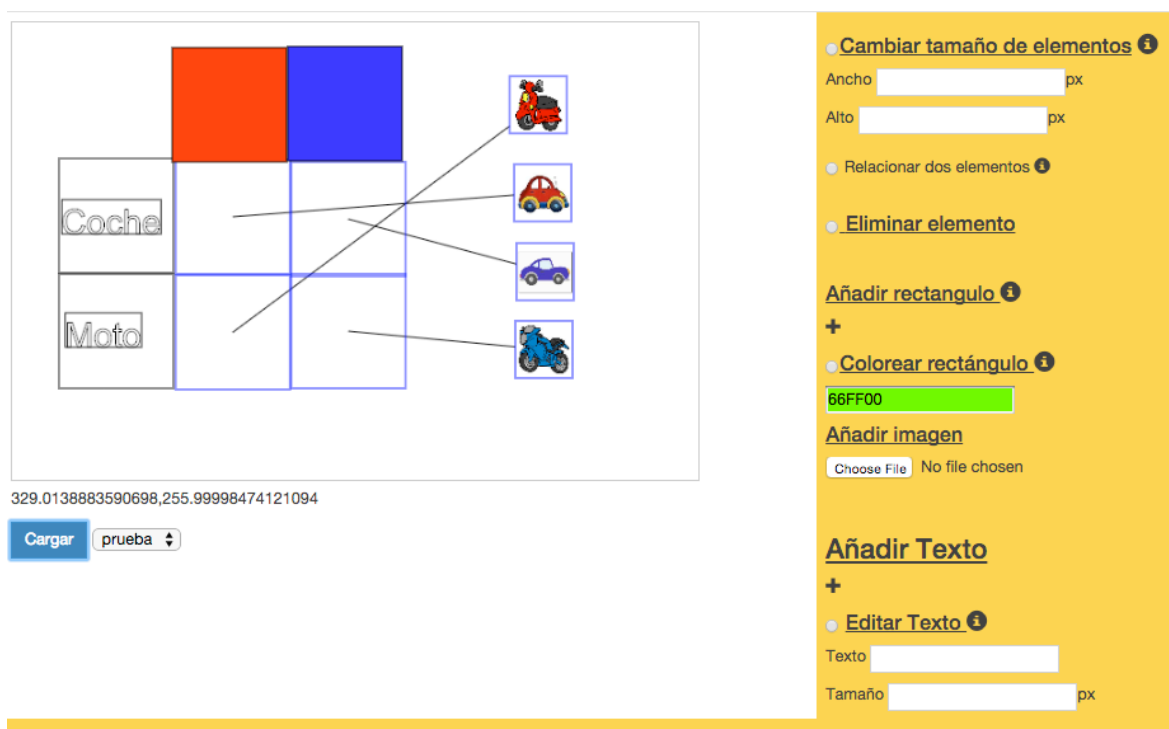


Figura A-10 Plantilla 4 de la aplicación

Esta plantilla te permite generar juegos donde el objetivo es mover unos elementos sobre otros o a ciertas zonas del juego si se usa un rectángulo para delimitar dicha zona.

Después de colocar los elementos del juego solo tienes que relacionar las parejas de elementos para que, cuando se ejecute el juego, si un elemento acaba lo suficientemente cerca de otro, suene el sonido de éxito.

Ejemplo

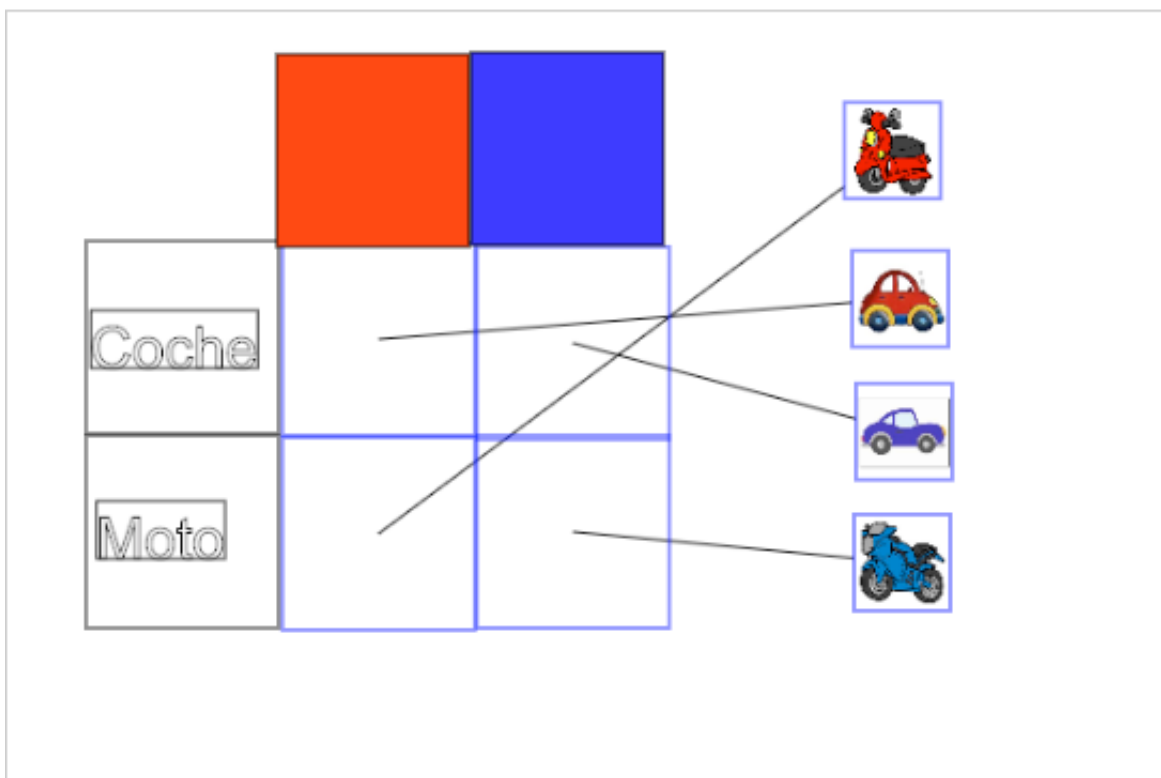


Figura A-11 Ejemplo de juego de plantilla 4

En este juego se ha improvisado una tabla usando rectángulos coloreando dos de ellos y añadiendo dos palabras, podemos dar a entender con ayuda de la instrucción "Coloca cada imagen en su sitio" Que queremos que se muevan los dibujos a sus sitios correspondientes en función de las propiedades de color y objeto en la imagen. Relacionando cada imagen con la celda correspondiente (que en realidad es un rectángulo) conseguimos que al mover las imágenes a las posiciones relacionadas suene el sonido de acierto.

Plantilla 5: Juegos de escribir palabras

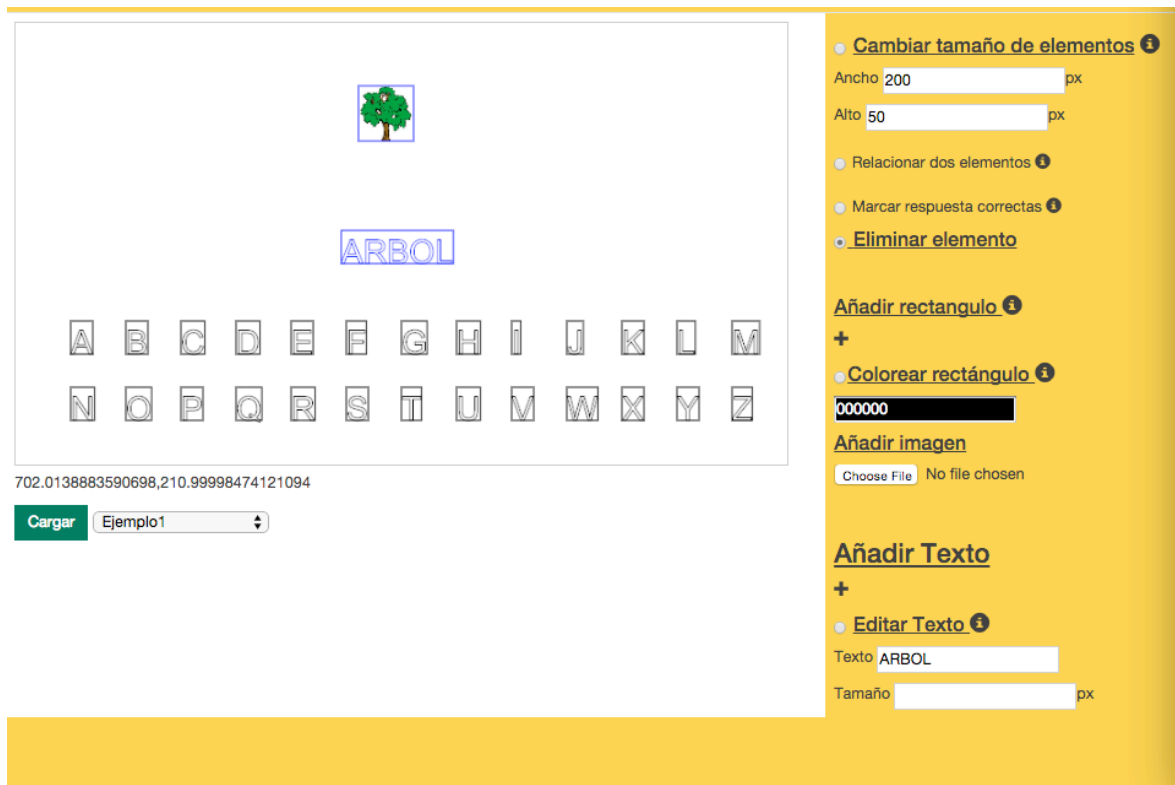


Figura A-12 Plantilla 5 de la aplicación

Esta plantilla permite generar juegos donde el objetivo es generar palabras correctamente para ello se pueden añadir distintas imágenes que sirvan de pista para saber que palabra escribir. Todos los textos que añadas aparecerán sustituidos por un rectángulo vacío. Para escribir la palabra pulsa el rectángulo y después las letras para escribir la palabra. Si la palabra coincide sonará la música de éxito.

Ejemplo

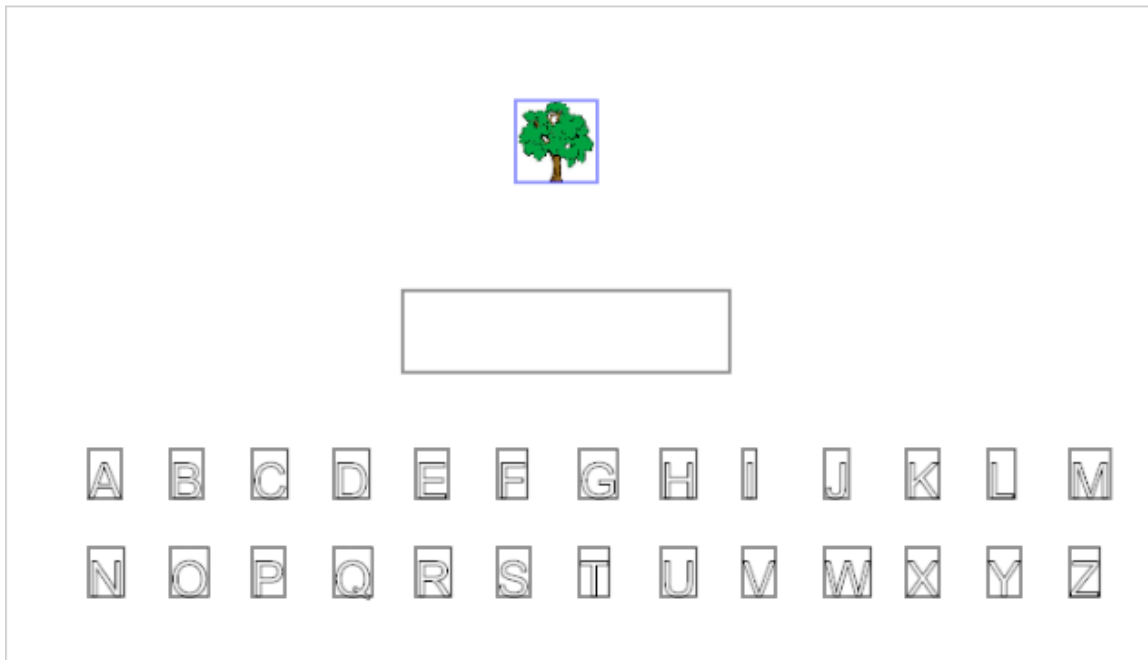


Figura A-13 Ejemplo de juego de plantilla 5

En este ejemplo se ha añadido una imagen y el texto árbol. Al cargar el juego la palabra es sustituida por un rectángulo vacío. Si después de pulsar el rectángulo pulsamos las letras para formar la palabra ‘árbol’ y volvemos a pulsar el rectángulo sonará el sonido de acierto, si nos equivocamos sonará el sonido de error y se borrará la palabra.

OPCIONES



Figura A-14 Opciones

En la sección de opciones dispones de las acciones para guardar y cargar juegos. Los juegos son almacenados en tu navegador, aunque lo cierres los datos guardados no se perderán

Con el botón *Borrar juegos* podrás borrar los juegos almacenados en el navegador.

Con el botón *Exportar Juegos* se descargará el archivo con los datos de los juegos creados hasta el momento y que están guardados en el navegador.

Usa esta opción cuando quieras guardar los juegos en el caso de que por ejemplo necesites reinstalar tu sistema operativo o tu navegador y no quisieras perder los datos.

Con la opción de *Importar juegos* podrás importar archivos con juegos que te hayan dado o hayas exportado anteriormente.

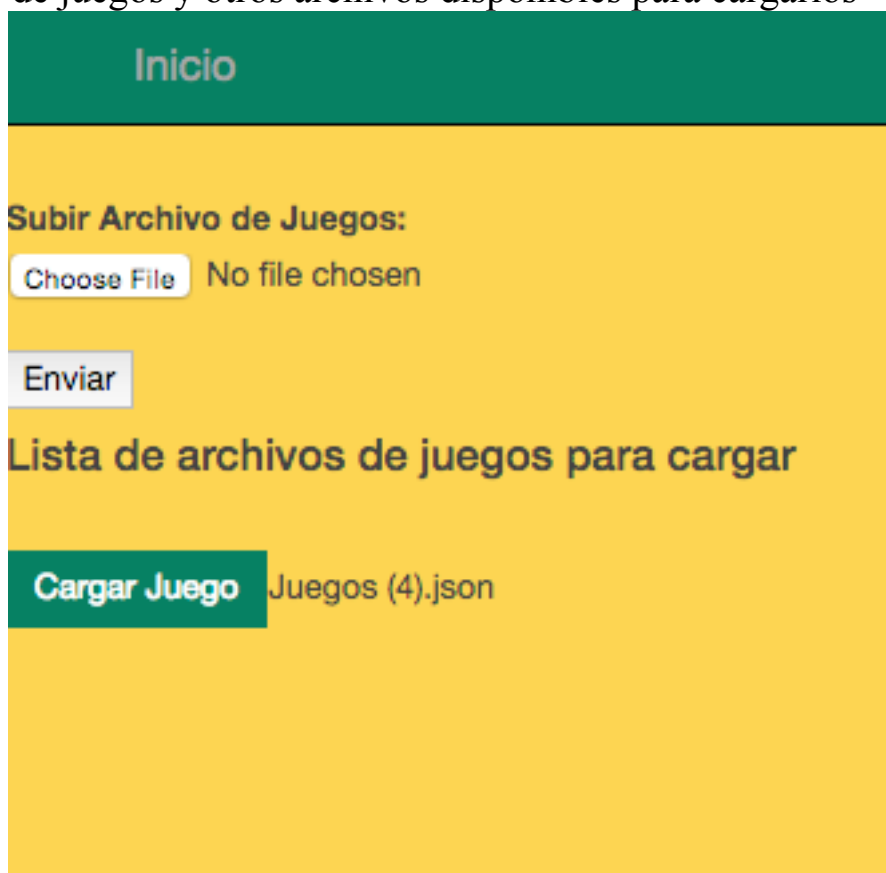
VERSION DE SERVIDOR

Para jugar desde una *tablet* es necesario hacer uso de la aplicación para servidor conectándose a donde se te indique que está instalada. Esta versión tiene una opción nueva: '*Carga de juegos*'



Figura A-15 Página de inicio en la version del servidor

Esta nueva opción te llevará una nueva página donde podrás subir archivos de juegos y otros archivos disponibles para cargarlos



The screenshot shows a web interface with a green header bar containing the word "Inicio". Below the header, on a yellow background, is the section "Subir Archivo de Juegos:". It features a "Choose File" button and the text "No file chosen". Below this is an "Enviar" button. Further down is the section "Lista de archivos de juegos para cargar", which contains a table with one row. The row has a green button labeled "Cargar Juego" and the text "Juegos (4).json".

Archivos
Cargar Juego Juegos (4).json

Figura A-16 Opciones para cargar y subir archivos en el servidor

Una vez seleccionado alguno de los archivos haciendo uso del botón correspondiente el archivo de juegos se cargará y se podrá jugar desde las páginas correspondientes.

B Informe de la coordinadora de Infantil.

VALORACION DE LA PRÁCTICA CON ALUMNOS DE 5 AÑOS.

En primer lugar comentaré que los juegos son divertidos, fáciles para los alumnos en cuanto a estímulo-respuesta.

También los alumnos han estado muy participativos interactuando de una manera ágil y rápida.

Referente al profesor me parece que con este sistema no te limitas a utilizar juegos que te proporcionan las editoriales sino que por el contrario puedes crear tus propias actividades en función de las necesidades del momento.

Lo ideal es tener este sistema creado para poder hacer un banco con juegos diferentes imágenes y datos.

Dicho sistema me parece fácil de hacer porque las indicaciones están bien explicadas y la manera de seguir los pasos es sencilla.

Tutora de 5 años.



